UNIVERSIDAD ANDINA NÉSTOR CÁCERES VELÁSQUEZ

FACULTAD DE INGENIERÍA DE SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA

PRESENTADO POR:
HUBER AGUILAR PINTO

Para Optar el Título Profesional de Ingeniero de Sistemas

JULIACA – PERU
2016

UNIVERSIDAD ANDINA NÉSTOR CÁCERES VELÁSQUEZ

FACULTAD DE INGENIERÍAS DE SISTEMAS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA "

TESIS PRESENTADO POR EL BACHILLER:

HUBER AGUILAR PINTO

PARA OPTAR EL TITULO PROFESIONAL DE:

"INGENIERO DE SISTEMAS"

APROBADO POR:

PRESIDENTE :

MSc. Juan Carlos Herrera MIranda

PRIMER MIEMBRO :

Dr. Rodolfo Fredy Arpasi Chura

SEGUNDO MIEMBRO :

MSc. Lucas Coaquira Cano

NÉSTOR CÁCERES VELÁSQUEZ

RESOLUCIÓN Nº 134-2016-D/FIS-UANCV

Juliaca, 2016 diciembre 26

VISTOS; El expediente Nº 18587, el oficio Nº 196-2016-D/EP-IS/FIS-UANCV-J de fecha 22 de diciembre del año 2016, del Presidente Jurado Dictaminador del Borrador de Tesis, **y el Acta de Aprobación de Borrador de Tesis de fecha 02 de diciembre del año 2016**, para optar el título profesional de Ingeniero de Sistemas, con el tema titulado: **"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA"**, presentado por el Bachiller: **HUBER AGUILAR PINTO,**

CONSIDERANDO:

Que, la (el) señor (ita), **HUBER AGUILAR PINTO**, ha presentado su Borrador de Tesis titulado: **"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA"**, para optar el Título Profesional de Ingeniero de Sistemas.

Que, habiendo procedido de acuerdo al Reglamento de Grados y Títulos de la UANCV, el Presidente de la Comisión de Grados y Títulos de la Facultad de Ingeniería de Sistemas, nominó como Jurados a los siguientes Docentes:

- Presidente : MSc. Juan Carlos Herrera Miranda
- 1er. Miembro : Dr. Rodolfo Fredy Arpasi Chura
- 2do. Miembro : MSc. Lucas Coaquira Cano

Que, el Jurado Dictaminador ha aprobado en su integridad el Borrador de Tesis titulado: **"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA,** y

Estando en la opinión favorable del Presidente de la Comisión de Grados y Títulos de la Facultad de Ingeniería de Sistemas, en concordancia al Reglamento de Grados y Títulos de la UANCV y en uso de las atribuciones que le concede la Ley Universitaria 30220, Ley de Creación de la UANCV 23738 y modificaciones, Resolución de Institucionalización 1287-92-ANR, D.L. 739 y el Estatuto de la UANCV.

SE RESUELVE:

ARTÍCULO PRIMERO.- APROBAR EL BORRADOR DE TESIS, para optar el Título Profesional de Ingeniero de Sistemas, con el tema titulado: **"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR EXAMENES EN LINEA"** presentado por el Bachiller: **HUBER AGUILAR PINTO,**

La misma que deberá proceder a desarrollar el temario de tesis aprobado de acuerdo a lo establecido en el Reglamento de Grados y Títulos de la UANCV.

ARTÍCULO SEGUNDO.- La Comisión de Grados y Títulos de la Facultad de Ingeniería de Sistemas y el Secretario Académico de la Facultad de Ingeniería de Sistemas, quedan encargados del cumplimiento de la presente Resolución.

Regístrese, Comuníquese y Archívese

UNIVERSIDAD ANDINA "NÉSTOR CACERES VELÁSQUEZ"
DECANO
Mgtr. Alcides Velásquez Ari
DECANO DE LA FIS

Distribución:
Interesado, Arch.
AVA/vipc

INDICE

CAPÍTULO I IDENTIFICACION DEL PROBLEMA

1 IDENTIFICACIÓN DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA

Actualmente los profesores en las diversas instituciones educativas al momento de realizar las pruebas de evaluación, no cuentas con una herramienta informática que les permita automatizar este proceso es por esto que se pretende con este proyecto solucionar este problema y crear una plataforma web que permita automatizar el proceso y que solucione esta deficiencia.

Los Docentes de nivel universitario tienen la necesidad de realizar evaluaciones periódicas para controlar el proceso de enseñanza aprendizaje, en sus estudiantes es por esto que se realizan muchas evaluaciones las cuales se pueden generar en forma automática mejorando el proceso de evaluación.

Entonces ¿es posible mejorar el proceso de evaluación empleando una herramienta que automáticamente permita crear exámenes objetivos de alternativa múltiple.

La generación de exámenes objetivos de alternativa múltiple se puede automatizar mediante el uso de Internet y lenguajes de software libre.

Gracias a al lenguaje PHP, se consigue crear páginas Web en tiempo real que provienen de unas plantillas (la implementación de las distintas plantillas será explicadas detalladamente para su mejor comprensión) y de la información existente en una Base de Datos, consiguiendo la creación de un sitio Web fácil de actualizar y consistente. Con esta tecnología la actualización del Web queda relegada a la simple administración de una Base de Datos, para la cual, se ha creado una sección de administración donde el administrador pueda llevar a cabo esta función de forma fácil, rápida e intuitiva.

En las Bases de Datos se almacena la información cambiante de cada página, a la que posteriormente accederemos para recuperar aquella información que se necesite para completar la plantilla para mostrársela al usuario en el momento que pida una página al servidor Web. Construyéndose el código HTML de una página u otra según la petición del usuario. Una vez cargada la página PHP solicitada por el usuario, a este lo único que le llamará la atención y por lo que notará que está visitando una página PHP, será la extensión de la misma en la dirección URL, que vendrá dada por las letras *php*, pero en lo referente al código HTML viene a ser igual que las páginas estáticas.

Como se ha dicho inicialmente, la intención es dar servicios on-line interesantes, novedosos y prácticos a los alumnos como pueden ser: la posibilidad de realizar test de diferente nivel de complejidad y saber la nota que ha obtenido al instante. Siguiendo además un estilo de diseño consistente para el sitio Web y sencillo de utilizar para que siempre se pueda saber en qué parte nos encontramos y no sé produzcan tantas perdidas como hemos observados en otras páginas.

Con lo anteriormente expuesto nos podemos platear la siguiente interrogante de investigación:

1.1.1 Problema General

¿Con un sistema automatizado que permita tomar exámenes en línea se agilizara este proceso mejorando la evaluación de estudiantes, facilitando esta labor a los docentes?

1.1.2 Problema Específico

1. Se podrá crear una herramienta informática mediante el empleo del ciclo de vida clásico para desarrollo de software.
2. Emplear software libre para abaratar los costos de desarrollo del proyecto de software.
3. Se utilizará las técnicas de la Ingeniería WEB para guiar y desarrollar el proyecto.

1.2 JUSTIFICACIÓN DE LA INVESTIGACION

Con el avance de la tecnología, surgen nuevos métodos que nos posibilitan automatizar los diferentes procesos educativos, uno de ellos es el proceso de evaluar a los estudiantes, tanto para los docentes como para los profesores, surge una necesidad de poder tener una herramienta que nos permita tomar exámenes (que son diseñados por los docentes) a los estudiantes en forma automática, lo cual hace que este proceso sea mejor y los resultados de estas evaluaciones sean obtenidos en el instante lo cual hace transparente este proceso.

Es por esto que se justifica el desarrollo del presente trabajo de investigación.

1.4 OBJETIVOS:

1.4.1 OBJETIVO GENERAL:

Elaborar un Sistema de software en la WEB con el fin de automatizar el proceso de elaborar, ejecutar y calificar un examen de alternativa de selección múltiple.

1.4.2 OBJETIVOS ESPECÍFICOS:

- Estudiar el ciclo de vida clásico para el desarrollo de software, así como emplearlo para la creación del sistema de software WEB.
- Utilizar el lenguaje de programación PHP y el sistema gestor de base de datos MySql para la creación del sistema WEB.
- Describir las técnicas de la Ingeniería WEB y emplearlas en la ejecución del sistema WEB.

1.5 HIPOTESIS

Mediante la implementación de una plataforma de software que genere y administre exámenes en línea se mejorara la evaluación de los estudiantes de una institución educativa.

1.5.1 Hipótesis especificas

1. Mediante el empleo de software libre es posible generar aplicaciones web de un buen nivel.
2. Con la utilización de La ingeniería del Software es posible la creación de un sistema web que cumpla con su propósito.
3. Empleando Ingeniería del Software se mejora la calidad de creación de software.

3.2 OPERACIONALIZACION DE VARIABLES

3.2.1 Variable Independiente

Aplicación WEB

Indicadores:

- Análisis

  Captura de requisitos Funcionales y no Funcionales

- Diseño

  Diseño de Interfaces, Base de Datos

- Implementación

  Codificación

- Pruebas

3.2.2 Variable Dependiente

- Exámenes en Línea

  Bueno

  Regular

  Malo

CAPÍTULO II MARCO TEÓRICO

2.1.- Antecedentes

SISTEMA DE GENERACION Y EVALUACION DE EXAMENES BASADO EN JAVA

Pedro J. Sánchez, Luis Martínez, Mª Dolores Muñoz

Departamento de Informática

Universidad de Jaén, 23071 – Jaén, España 1999

SISTEMA DE EVALUACIONES A TRAVES DE INTERNET

Enzo Geovanny Gómez Fernández, Rene Javier Conde Yanme

Licenciatura en Computación

El Salvador 2005.

2.1.2.- Los lenguajes de Internet

Como todo gran invento, Internet también tiene su secreto. La creación de un estándar de intercambio llamado HTTP, y de un lenguaje descriptivo llamado HTML, utilizado para la creación de páginas web.

Dicho lenguaje, de dominio público y fácil aprendizaje, fue el primer causante del *boom* de creadores y diseñadores de páginas HTML, los cuales, con un simple editor de texto y un poco de espacio de disco duro en un servidor web, contribuían a forjar la infinita información que está disponible desde la Web. En Internet la información se divide en cientos de miles de temas, en todos los formatos imaginables. Tiene a su disposición todo tipo de documentos de texto, imágenes, sonidos, vídeos, charlas on-line sobre cualquier tema.

Todos estos contenidos se crean desde la base del HTML, pero este lenguaje siempre se ha caracterizado por ser un lenguaje estático, es decir, una vez cargada la página en nuestro navegador (navegador es el programa que nos permite visualizar las páginas HTML, tales como, Netscape Navigator, Mozilla Firefox y Microsoft Internet Explorer) esta no puede ser modificada ni en la forma ni en el contenido. Para remediar esto surgen los *lenguajes de script*, que nos permiten realizar ciertas operaciones on-line en la parte cliente (el navegador).

- **Los lenguajes de script**

Los lenguajes de script tales como, *Visual Basic Script* o *Java Script*, se ejecutan sin necesidad de compilarlos, como apoyo a otros programas mayores, y siempre dentro de una aplicación cliente, el navegador. Son como una extensión al lenguaje descriptivo HTML, para ampliar su funcionalidad.

Pero no queda todo aquí, ya que los lenguajes de script, que se ejecutan en la parte cliente a pesar de dar mayor funcionalidad a las especificaciones del HTML, no puede ofrecer una interactividad entre el cliente y el servidor web, hacen falta de aplicaciones que se puedan ejecutar en el servidor para poder ofrecer al usuario páginas dinámicas, creadas en tiempo real, en función de la petición del cliente, y al mismo tiempo poder procesar y almacenar la información, así aparecen *las aplicaciones CGI*, que son aplicaciones que se ejecutan en el lado del servidor.

- **El CGI (*Common Gateway Interface)***

Es un estándar para comunicar aplicaciones externas con los servidores de información. Un documento en HTML que el navegador se trae es estático, un programa CGI, por otro lado, es ejecutado en tiempo real, así que puede generar información dinámica, dependiendo de los datos enviados por el cliente.

El comportamiento dinámico es uno de los puntos más importantes de la Red y a menudo es difícil de apreciar. Esto se evidencia sobre todo cuando se crean páginas web sobre la marcha en respuesta a las peticiones de los usuarios, aunque lo que le aparezca al usuario que sea igual que otra página web más, siendo en realidad un documento temporal que se ha creado sólo para dicha petición.

CGI proporciona el medio de conexión entre los servidores web y las fuentes de información. Los scripts o los programas CGI se pueden considerar como extensiones del núcleo de un servidor web, las aplicaciones CGI procesan información, y realizan tareas de recuperación y formateo para los servidores. La interfaz CGI define un método para el servidor web, para que este acomode programas y servicios adicionales que se utilizarán para acceder a aplicaciones externas desde dentro del texto de una página web.

El término *gateway* describe la relación que hay entre el servidor web y las aplicaciones externas que se encargan de acceder a los datos y de manipular tares. Estas *puertas de acceso* manipulan cierta información y devuelve una respuesta relacionada con la petición recibida, por ejemplo, un documento HTML (página web) generado sobre la marcha, que incluye los resultados de una petición a una base de datos externa. En otras palabras, el CGI permite que el servidor les suministre información a los clientes, los cuales no podrán acceder a ella por cualquier otro camino.

Por ejemplo, supongamos que quieres enganchar una base de datos al Web, para permitir a gente de todo el mundo consultarla. Básicamente se necesitará un programa CGI que el Servidor ejecutará para transmitir la información al gestor de base de datos, y recibir los resultados para presentárselos al cliente. Este es un ejemplo de pasarela (*gateway*).

El ejemplo de la base de datos es una idea sencilla, pero la mayoría de las veces difícil de implementar. Realmente no hay límite con lo que tú quieras enganchar al Web. La única cosa que debes recordar es que haga lo que haga tu programa CGI, no deberá tardar mucho tiempo en procesar. De otra manera, el usuario estaría esperando con su navegador a que algo pase.

En resumen, CGI permite el intercambio de información y servicios que están fuera del alcance de la Web produciendo objetos HTTP para aquellos que no lo son, permitiendo que el usuario pueda verlos.

- **Ventajas de la Tecnología PHP sobre el CGI**

La programación CGI tradicional, utilizada para crear páginas web dinámicas, y realizada normalmente en lenguajes como Perl o C++, tiene muchos inconvenientes.

La mayoría de las veces el CGI se basa en un programa ejecutable, que como todos los ejecutables tienen que ser compilados y linkados, haciendo de la programación CGI un proceso lento y laborioso. El desarrollo de aplicaciones en estos casos se hace un proceso lento y tedioso, con múltiples *bugs* y dificultades.

En otros casos, el lenguaje elegido suponía un esfuerzo de aprendizaje adicional, y no siempre estaban bien resueltos aspectos tales como control del sistema, el acceso a bases de datos, etc.

La programación PHP soluciona estos problemas, al no tener que invertir un gran esfuerzo en aprendizaje (su sintaxis es muy similar a la de C, C++, Java y Perl), y al tratarse de un lenguaje interpretado no hace falta consumir tiempo en compilarlo, y hace que la tarea de corrección de errores sea más ágil y dinámica.

Por otro lado, ofrece una potencia y un control sobre el sistema que hasta ahora no daba ningún otro lenguaje de programación CGI. Además, el acceso a bases de datos es extremadamente sencillo, y totalmente compatible con muchas bases como pueden ser Adabas Ds, dBae, Empress, FiclePro, informix, Internase, Solid, Sybase, Veloces, Unix dbm, mSQL, MySQL, Oracle, PosgreSQL, etc.

2.1.3.- El HTML (HyperText Markup Language)

Una página web o HTML es una página de texto en la que se incluyen una serie de identificadores o etiquetas descriptivas, que ayudan a definir zonas, formatos y características del texto. Por lo tanto, las páginas web son únicamente ficheros de texto que incluyen una serie de identificadores que hacen que tenga un formato en concreto.

El programador o diseñador escribe el código de la página utilizando las etiquetas descriptivas correspondientes, pudiendo insertar elementos multimedia, tales como, imágenes, vídeo y sonido. Luego son los navegadores los que se encargan de traducir el código HTML en un formato gráfico, mostrando el resultado final en la pantalla del ordenador.

Formalmente el HTML es una aplicación del SGML (Standard Generalized Markup Language). En 1986 se aprobó el SGML como un estándar internacional, el cual, proporciona una forma de representación de la estructura de documentos y de hiperdocumentos. También es una manera de codificación de hiperdocumentos que permite su intercambio.

El SGML es un *metalenguaje* centrado en la descripción formal de los sistemas de referencia de documentos. Resumiendo, HTML usa el SGML como un lenguaje que describe la estructura de un hiperdocumento y su interconectividad.

El HTML tiene cuatro características principales:

- Utiliza etiquetas descriptivas.
- Sigue una estructura jerarquizada, además de enlaces con elementos del documento y ajenos a él.
- El HTML se rige por una especificación formal el DTD (Document Type Definition), que define la sintaxis del HTML.
- La especificación HTML, además del propio lenguaje, es legible por los ordenadores y por los humanos.

- **Etiquetas descriptivas**

El HTML es un lenguaje que se basa en etiquetas descriptivas. Esto quiere decir que nos vamos a encontrar partes del documento que están marcadas o etiquetadas con nombres que las describen. Las etiquetas le identifican al lector que ciertas frases o palabras son especiales.

Las etiquetas descriptivas tienen la siguiente sintaxis:

<etiqueta>:	Se utiliza como marca de comienzo de formato, es decir, todo lo que vaya a continuación de dicha etiqueta se le aplicará un formato especial.
</etiqueta>:	Indica que se quiere acabar el formato que antes se había comenzado.

- **Estructura jerarquizada e interconexiones**

Los enlaces entre documentos o entre elementos de un mismo documento están ligados a la estructura de un documento HTML. Una interconexión o enlace es una relación entre dos elementos de documentos. Estos elementos se llaman *puntos de enlace*.

El punto de enlace origen está enlazado con el punto de enlace final. En HTML, los puntos de enlace pueden tener nombres o no, a través de la asignación de etiquetas de enlace.

Los elementos de un documento pueden ser generales, como una frase, o específicos, como una palabra o píxel. Los elementos de los documentos pueden hacer referencia a otros elementos o a objetos externos a través de un vínculo de un hipertexto, de acuerdo con la disciplina que se esté utilizando en dicho documento.

Con esto tenemos que en HTML los vínculos pueden enlazar elementos de un mismo documento (vinculación intra-documento) o con elementos de otros documentos (vinculación inter-documentos).

- **Especificación formal**

El HTML tiene un DTD (Document Type Definition) que define su especificación formal. El DTD define la sintaxis del HTML, describe cada elemento de un documento HTML, define los atributos permitidos para cada elemento y describe el modelo de contenido de datos de cada uno de ellos (un modelo de contenido de datos define que elementos pueden operar a la vez). Además de la información del elemento, el DTD provee definiciones para cualquier entidad externa a la que se pueda hacer referencia desde un documento HTML (por ejemplo, los caracteres IOS-Latin-1 utilizados para representar caracteres en la pantalla).

A lo largo de la historia, el HTML DTD ha pasado por tres niveles, actualmente es la especificación de HTML que admite la IETF (Internet Engineering Task Force).

- **Representación legible**

El HTML representa la estructura de un documento de forma que sea legible por personas y ordenadores. Las etiquetas (la descripción que ayuda a definir el HTML y su comportamiento) se encuentran separadas del texto por una serie de cadenas limitadoras, las cuales son imprimibles; texto y etiquetas pueden coexistir en paz.

Los nombres de las etiquetas utilizan una terminología que puede ser comprendida por la mayoría de las personas (o abreviaturas de dicha terminología, que cobran sentido cuando se explican).

2.2.- Tecnología PHP (Hypertext Preprocessor)

2.2.1.- ¿Por qué PHP?

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Esta nueva tecnología creada por Rasmus Lerdof a finales de 1994 permite diseñar completas aplicaciones que permiten flexibilizar las tareas de acceso y procesado de los datos en el servidor utilizando tan sólo HTML y un lenguaje de script convencional. Ahora el código Script se puede ejecutar en el servidor y no es necesario que lo haga en la máquina cliente.

Una respuesta corta y concisa, pero ¿qué significa realmente? Un ejemplo nos aclarará las cosas:

Ejemplo 2.1. Un ejemplo introductorio

```
<html>
<head>
<title>Ejemplo PHP</title>
</head>
<body>
<?php echo "Hola, este es un ejemplo con PHP!"; ?>
</body>
</html>
```

Podemos ver que no es lo mismo que un script CGI escrito en otro lenguaje de programación como Perl o C -- En vez de escribir un programa con muchos comandos para crear una salida en HTML, escribimos el código HTML con cierto código PHP embebido (introducido) en el mismo, que producirá cierta salida (en nuestro ejemplo, producir un texto). El código PHP se incluye entre etiquetas especiales de comienzo y final que nos permitirán entrar y salir del modo PHP.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los ficheros HTML con PHP.

Con PHP el desarrollo de contenido Web en ambos lados del modelo Cliente/Servidor se convierte en un mismo trabajo. No hay necesidad de aprender nuevas herramientas de desarrollo.

Las Ventajas proporcionadas por PHP frente a otros conceptos de desarrollo en el Servidor son:

- Su simplicidad: se les debe permitir a los programadores generar código en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente, debido a que sencillamente es un lenguaje de scripting embebido en páginas web, con una serie de librerías accesorias de fácil manejo y completa funcionalidad que abarca prácticamente todo lo necesario e imaginable para desarrollar aplicaciones web de complejidad baja- media. Es sencillo de aprender, así como de programar y está bastante extendido.

- Conectividad: la conectividad se puede llevar a cabo de forma sencilla mediante conjuntos de librerías.

- Velocidad: no solo la velocidad de ejecución es importante, sino además no crear demoras en la máquina. Por esta razón un buen lenguaje dinámico no debe requerir demasiados recursos de sistema. Estos requisitos los cumple PHP, el cual se integra muy bien junto a otros softwares, especialmente bajo ambientes Unix y sobre todo cuando se configura como módulo de Apache.

- Estabilidad: la velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

- Seguridad: el sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.

- Su extensibilidad. Las páginas PHP soportan el uso de cualquier lenguaje de Script (que cumpla las convenciones de ActiveX Scripting) y el uso de componentes desarrollados en cualquier lenguaje (Visual Basic, C++, Java, Delphi...). Si necesita alguna característica adicional sólo tiene que programar un componente y automatizarlo desde una página PHP. Se proporciona un completo modelo de objetos para ayudar a los programadores de componentes a comunicarse con PHP.

- Proporcionan un rápido acceso a las variables de entorno del servidor y permiten guardar información entre diferentes llamadas.
- Permite de una manera sencilla y Rápida el acceso a bases de datos, obtención de información del usuario que accede, etc.

- Debido a que PHP trabaja en el mismo espacio de procesos que el servidor y, por consiguiente, es mucho más rápido y está más optimizado que los script o aplicaciones CGI.

- Permiten proceso en páginas web con unas importantes diferencias respecto a los scripts en el lado del cliente (navegador), y es que al ejecutarse en el servidor y devolver sólo HTML se podrán utilizar desde cualquier tipo de navegador sin tener que estar ligado a un navegador en concreto.

Hasta la llegada de la programación de páginas PHP el sistema más utilizado para enviar y recibir datos a través de Internet era el método conocido como CGI. Es decir, programas que están instalados en el servidor. Fundamentalmente este sistema tenía tres objetivos: uno, la recepción y lectura de datos provenientes de un formulario web en el que el usuario introducía la información que mandaba al servidor; dos, procesar esa información y acceder a otros sistemas, por ejemplo, bases de datos; y tres, devolver los resultados a través de Internet para llegar al navegador del usuario en formato HTML.

Cuando la información se hace realmente grande en un servidor, o bien, se tiene el acceso a bases de datos como motor principal, suele resultar mucho más rentable gestionar los datos mediante un motor de bases de datos y presentárselos al usuario como plantillas tipo HTML.

Las páginas PHP vienen a dar el último paso en lo que a programación Internet se refiere. El ahorro de tiempo con respecto a la programación tradicional CGI es enorme, y la reutilización que podemos hacer de nuestro código es lo suficientemente alta como para hacer de PHP una de las tecnologías más a tener en cuenta.

1.7 CRONOGRAMA DE ACTIVIDADES

Fases/ Meses	**JUNIO**				**JULIO**				**AGOSTO**				**SETIEMBRE**			
Fase Inicio(5 Semanas)	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**		**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
ACTIVIDADES																	
Descripción del problema.																	
Modelo del negocio.																	
Lista de requerimientos.																	
Casos de uso del sistema.																	
Objetivos del sistema de información.																	
Estudio de factibilidad.																	
Análisis de riesgos.																	
Fase elaboración																	
Diseño de los casos de uso Funcionales																	
Diagrama de Clases																	
Diagrama de Colaboración																	
Prototipo																	

Arquitectura Candidata																	
Hardware y software, Empleado en el desarrollo del sistema																	
Fase Transición (1 semana)																	
Pruebas																	
Manual de Usuario																	

CAPÍTULO III METODOLOGIA DE DESARROLLO DEL SISTEMA WEB

3.- Diseño del Sistema Ciclo de Vida Cascada

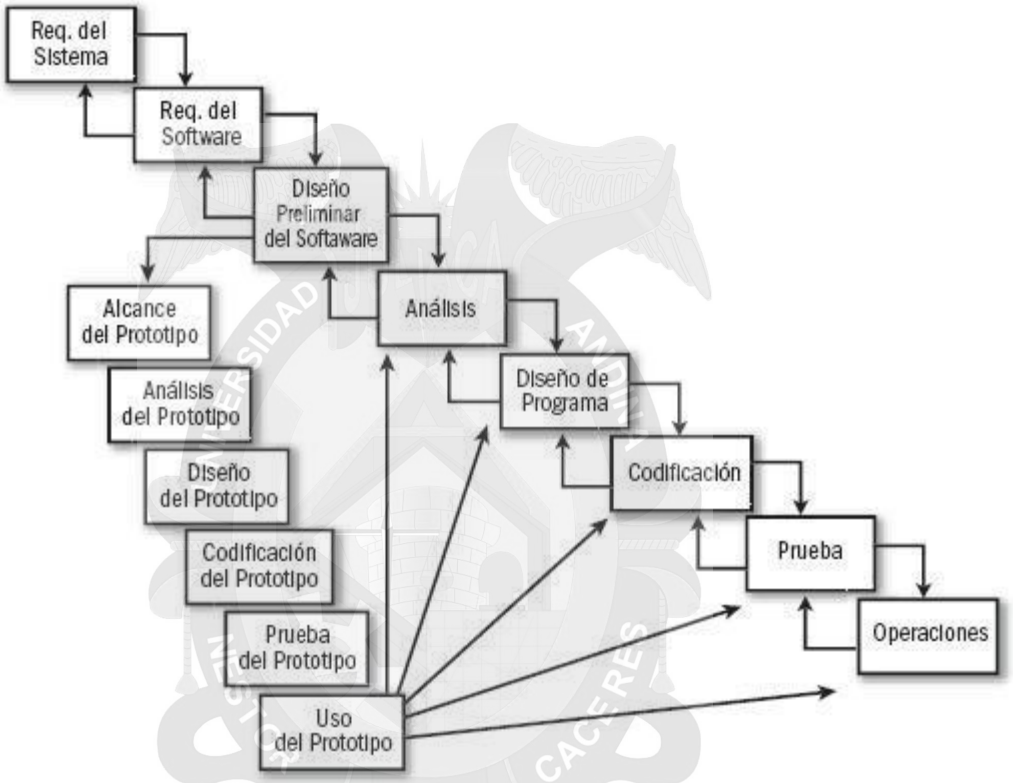


Figura 3.1 Ciclo de Vida en Cascada

En un modelo en cascada un proyecto progresa a través de una secuencia ordenada de pasos que son:

Concepto del software.

- Análisis de requerimientos.
- Diseño global.
- Diseño detallado.
- Codificación y depuración.
- Prueba del sistema.

El modelo contiene una serie de etapas que no se solapan, y el proyecto se va revisando tras cada una de las etapas. Para poder pasar a la siguiente etapa se tiene que haber conseguido todos los objetivos de la etapa anterior, es un proceso secuencial.

Tiene una buena aplicación cuando el problema es estable y cuando se trabaja con metodologías técnicas conocidas. Este modelo será apropiado para la migración de una aplicación o a una versión de mantenimiento bien definida.

Con este modelo se tiene un seguimiento de todas las fases del proyecto y del cumplimiento de todos los objetivos marcados en cada etapa tanto de costos, fechas de entrega y lo más importante que pueden comprobar al final de cada etapa si el proyecto cumple todas las necesidades del usuario.

A su vez esto es un problema ya que, si el usuario se da cuenta de que falta una tarea de la empresa en el proyecto una vez pasada esta etapa, el trabajo que hay que realizar se retrasa en fechas de entrega y el costo es mayor. Por lo tanto, esto produce un fracaso en la industria ya que es reacio a las modificaciones de última hora.

Por este motivo se puede modificar el modelo en cascada pudiendo pasar de una etapa a la anterior, pero suele ser difícil ya que hay que rehacer la etapa anterior, este modelo es el ciclo de vida del salmón. Por lo tanto, este es un modelo poco apropiado para proyectos con fecha de entrega corta, pero su rendimiento puede mejorar notablemente variando el modelo de la cascada pura.

3.1.-Planificación Temporal

Trabajo: Proyecto
Nombre del trabajo:. **Descripción del trabajo:** . • Realizar una herramienta flexible, eficaz, fácil e intuitiva de manipulación de datos. • Dar de alta una serie de alumnos en el sistema y que estos puedan realizar una batería de test, de forma que sea un apoyo para su aprendizaje. • El administrador se encargará de la gestión y mantenimiento de toda la información del Nodo Web.
Recursos: Propios, pc's, tarjetas de red, acceso a Internet **Personal:** Administrador, usuarios y responsable del proyecto.
Requisitos: Disponer de la propuesta inicial del cliente. **Productos:** PC's nuevos. Red Local Ampliada. Servidor. Documentación del proyecto.
Tareas predecesoras: -Buscar documentación, bajarse herramientas de SW libre **Tareas sucesoras:** -Pruebas de ejecución, accesibilidad
Riesgos: No entender la propuesta del cliente. Disponer de hardware defectuoso.

Número de trabajo: 1	
Nombre del trabajo:	**Documentación y adquisición del SW libre**
Descripción del trabajo:	Generar un documento que especifique los recursos hardware necesarios para realizar la aplicación Web Así como gestionar, entrevistas necesarias con el Cliente
Duración estimada:	10 días
Requisitos:	Disponer de la propuesta inicial del cliente.
Productos:	Documento adquirir e instalar el Software nuevo. PC's Nuevos.
Tareas predecesoras:	-
Tareas sucesoras:	
Riesgos:	No disponer de toda la información necesaria. Los PC's Son defectuosos.

Número de trabajo: 2	
Nombre del trabajo:	**Desarrollar e Instalar el Software Nuevo**
Descripción del trabajo:	Generar documentación con la especificación de Instalación así como instalar dicho software nuevo. Aprender su funcionamiento: MySQL, servidor Apache, PHP
Duración estimada:	5 días
Recursos:	Propios.
Personal:	Técnicos y responsable de proyecto.

Requisitos:	Disponer de la propuesta inicial del cliente. Disponer de servidor local (Apache).
Productos:	Documento del desarrollo e instalación del software nuevo. Software nuevo (desarrollado e instalado).
Tareas predecesoras:	-1
Tareas sucesoras:	-
Riesgos:	No disponer de toda la información necesaria. Disponer de un hardware que no cumple los requisitos necesarios para la Instalación del software.

Número de trabajo: 3	
Nombre del trabajo:	**Planificación**
Descripción del trabajo:	Especificar los requisitos de la aplicación. Realizar un estudio de la viabilidad de la aplicación. Establecimiento de los objetivos y generación de Documentación de planificación.
Duración estimada:	10 días
Recursos:	Propios
Personal:	responsable de proyecto.
Requisitos:	Disponer de la propuesta inicial del cliente.
Productos:	Documento de planificación.
Tareas predecesoras:	-
Tareas sucesoras:	-
Riesgos:	No disponer de toda la información necesaria

Número de trabajo: 3.0	
Nombre del trabajo:	**Desarrollar Base Datos.**
Descripción del trabajo:	Realizar base de datos. Generar documentación de desarrollo de base de datos.
Duración estimada:	15 días
Recursos:	Propios
Personal:	Cliente, técnicos y responsable de proyecto.
Requisitos:	Disponer de la propuesta inicial del cliente.
Productos:	Documento desarrollar base de datos. Base de datos.
Tareas predecesoras:	-
Tareas sucesoras:	3.1, 3.2, 3.3

Número de trabajo: 3.1.	
Nombre del trabajo:	**Diseño interno**
Descripción del trabajo:	Describir la estructura física de la BD.
Duración estimada:	10 días
Recursos:	Propios.
Personal:	Cliente, usuario, responsable de proyectos y técnicos.
Requisitos:	Disponer de la propuesta inicial del cliente.
Productos:	Diseño interno y documento de diseño interno.
Tareas predecesoras:	-
Tareas sucesoras:	3.2
Riesgos:	No disponer de la información necesaria por parte del cliente

Número de trabajo: 3.2	
Nombre del trabajo:	**Diseño conceptual**
Descripción del trabajo:	Describir la estructura de la BD en su totalidad.
Duración estimada:	15 días
Recursos:	Propios.
Personal:	Cliente, usuario, responsable de proyectos y técnicos.
Requisitos:	Disponer de la propuesta inicial del cliente y diseño interno.
Productos:	Diseño conceptual y documento de diseño conceptual.
Tareas predecesoras:	-
Tareas sucesoras:	3.3
Riesgos:	No disponer de la información necesaria por parte del cliente

Número de trabajo: 3.3	
Nombre del trabajo:	**Diseño externo**
Descripción del trabajo:	Describe la parte concreta de la BD para cada grupo de usuarios.
Duración estimada:	5 días
Recursos:	Propios
Personal:	Cliente, usuario, responsable de proyectos y técnicos.
Requisitos:	Disponer de la propuesta inicial del cliente y diseño conceptual.
Productos:	Diseño externo y documento de diseño externo.

Tareas predecesoras:	3.2
Tareas sucesoras:	-
Riesgos:	No disponer de la información necesaria por parte del Cliente

Número de trabajo: 4	
Nombre del trabajo:	**Análisis**
Descripción del trabajo:	Realizar un estudio de la función, comportamiento y rendimiento de la aplicación.
Duración estimada:	10 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer del documento de planificación.
Productos:	Documento de análisis del sistema.
Tareas predecesoras:	
Tareas sucesoras:	4.1
Riesgos:	El documento de planificación es erróneo o incompleto.

Número de trabajo: 4.1	
Nombre del trabajo:	**Diseño.**
Descripción del trabajo:	Generar un documento que especifique la arquitectura del software, la definición de las estructuras de datos a implementar, los detalles de las interfaces y los detalles procedimentales de la aplicación.
Duración estimada:	10 días
Recursos:	Propios.

Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer del documento de análisis del sistema y de la Base de Datos.
Productos:	Documento de diseño del sistema.
Tareas predecesoras:	4
Tareas sucesoras:	4.2, 4.3, 4.4
Riesgos:	El documento de análisis es erróneo o incompleto.

Número de trabajo: 4.2	
Nombre del trabajo:	**Codificación**
Descripción del trabajo:	Generar documentación que englobe los detalles de implementación del sistema.
Duración estimada:	70 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer del documento de diseño del sistema y de la Base de Datos.
Productos:	Codificación y documento de codificación.
Tareas predecesoras:	
Tareas sucesoras:	4.3
Riesgos:	El documento de diseño es erróneo o incompleto.

Número de trabajo: 4.3	
Nombre del trabajo:	**Pruebas**
Descripción del trabajo:	Realiza una serie de test técnicos, operativos y de instalación sobre el sistema

implementado. **Duración estimada:**	5 días
Recursos: **Personal:**	Propios Técnicos y responsable de proyecto.
Requisitos: del sistema. **Productos:** sistema.	Disponer de la codificación, del documento de codificación y el documento de diseño La aplicación y documento de pruebas del
Tareas predecesoras: **Tareas sucesoras:**	4.2
Riesgos:	Disponer de una codificación errónea.

Número de trabajo: 4.4	
Nombre del trabajo: **Descripción del trabajo:** respecto al **Duración estimada:**	**Mantenimiento** Realizar un estudio de la línea de trabajo mantenimiento de la aplicación construida. A estimar
Recursos: **Personal:**	Propios Técnicos.
Requisitos: **Productos:**	Disponer del documento de diseño de pruebas del sistema y de la aplicación -
Tareas predecesoras: **Tareas sucesoras:**	4.3 -
Riesgos: pedidos debidamente	No disponer de la aplicación gestión de Realizada.

Número de trabajo: 5	
Nombre del trabajo:	**Migración de datos**
Descripción del trabajo:	Preparar el entorno de migración y carga inicial de datos, así como la generación del código de los componentes y procedimientos de migración y carga inicial de datos.
Duración estimada:	12 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer de las aplicaciones instaladas. Disponer de la red local ampliada.
Productos:	Documento de migración de datos. Software nuevo instalado.
Tareas predecesoras:	5.1 y 5.2
Tareas sucesoras:	-
Riesgos	

Número de trabajo: 5.1	
Nombre del trabajo:	**Definición de objetivos**
Descripción del trabajo:	Especificar y definir los puntos u objetivos que debe englobar y cumplir el sistema.
Duración estimada:	13 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer del documento de requisitos.
Productos:	Documento de definición de objetivos.
Tareas predecesoras:	
Tareas sucesoras:	5.2
Riesgos:	No disponer del documento de requisitos.

Número de trabajo: 5.2	
Nombre del trabajo:	**Modelado dinámico**
Descripción del trabajo:	Obtener el modelo de procesos de la aplicación. Especificar las interfaces de usuario.
Duración estimada:	20 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	
Productos:	Documento de Modelo Dinámico.
Tareas predecesoras:	
Tareas sucesoras:	
Riesgos:	No disponer del documento de Casos de Uso.

Número de trabajo: 5.3	
Nombre del trabajo:	**Arquitectura del software**
Descripción del trabajo:	Definición de niveles de la arquitectura software: diagrama de despliegue. Identificación de requisitos de diseño y construcción: lenguajes.
Duración estimada:	3 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto
Requisitos:	Disponer del documento de Análisis del Sistema de Gestión de Pedidos y de la Base de Datos.
Productos:	Documento de Arquitectura Software.
Tareas predecesoras:	-
Tareas sucesoras:	
Riesgos:	No disponer del documento de Análisis del Sistema

Número de trabajo: 5.4	
Nombre del trabajo:	**Definición de las estructuras de datos**
Descripción del trabajo:	Definir las estructuras de datos del sistema
Duración estimada:	20 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto
Requisitos :	Disponer del documento Arquitectura Software y Análisis del Sistema.
Productos:	Documento Definición Estructuras de Datos
Tareas predecesoras:	
Tareas sucesoras:	
Riesgos:	No disponer de la información necesaria.

Número de trabajo: 5.5	
Nombre del trabajo:	**Detalle de la interfaz**
Descripción del trabajo:	Definir la interfaz de usuario. y dinámico realizado en la etapa de análisis.
Duración estimada:	3 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Documento de Análisis del Sistema y el documento de definición de Estructura de Datos.
Productos:	Documento de Detalle de Interfaz.
Tareas predecesoras:	
Tareas sucesoras:	-
Riesgos:	No disponer de la información necesaria.

Número de trabajo: 5.6	
Nombre del trabajo:	**Integración de componentes**
Descripción del trabajo:	Unificar e integrar los componentes y procedimientos del sistema.
Duración estimada:	1 días
Recursos:	Propios.
Personal:	Técnicos y responsable de proyecto.
Requisitos :	Disponer del documento del Código de componentes y procedimientos.
Productos:	Codificación de Gestión de Pedidos.
Tareas predecesoras:	
Tareas sucesoras:	
Riesgos:	No disponer de la información necesaria.

Número de trabajo: 5.6	
Nombre del trabajo:	**Elaboración del documento de codificación y manuales de usuario.**
Descripción del trabajo:	Realizar un documento que englobe toda la información analizada en la etapa de planificación.
Duración estimada:	4días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer de la codificación
Productos:	Documento Manuales de Usuarios
Tareas predecesoras:	-
Tareas sucesoras:	-
Riesgos:	La codificación es errónea.

Número de trabajo: 6	
Nombre del trabajo:	**Técnicas**
Descripción del trabajo:	Pruebas sobre elementos hardware, conexionado de equipo, etc.
Duración estimada:	3 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto
Requisitos :	Disponer del documento de diseño del sistema, documento de codificación y codificación.
Productos:	Documento de pruebas técnicas.
Tareas predecesoras:	-
Tareas sucesoras:	-
Riesgos:	No disponer de la información necesaria.

Número de trabajo: 7	
Nombre del trabajo:	**Operativas**
Descripción del trabajo:	Pruebas sobre la funcionalidad del sistema.
Duración estimada:	3 días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto
Requisitos:	Disponer del documento de Diseño del Sistema, documento de Pruebas Técnicas, documento de codificación, codificación
Productos:	Documento de Pruebas Operativas.
Tareas predecesoras:	-
Tareas sucesoras:	-
Riesgos:	No disponer de la información necesaria.

Número de trabajo: 8	
Nombre del trabajo:	**Instalación**
Descripción del trabajo:	Pruebas durante la instalación del sistema.
Duración estimada:	1días
Recursos:	Propios
Personal:	Técnicos y responsable de proyecto.
Requisitos:	Disponer de codificación, documento de pruebas operativas.
Productos:	Documento de Pruebas de Instalación y la aplicación de Gestión de Test.
Tareas predecesoras:	-
Tareas sucesoras:	-
Riesgos:	No disponer de la información necesaria.

CAPITULO IV ANÁLISIS Y RESULTADOS DEL SISTEMAS

4.1.- Análisis de las Bases de Datos.

Las Bases de Datos en sí pueden considerarse como una especie de archivador electrónico; dicho de otra manera, es un lugar donde se almacena un conjunto de registros de datos computarizados, los cuales, vamos a explicar a continuación, para cada una de las Bases de Datos.

A continuación, se pasa a informar de la Base de Datos que he considerado conveniente incluir en el Proyecto:

- Base de Datos *generacionexamenes*: donde se guardará toda la información referente a los usuarios (alumnos y administradores), test, preguntas, respuestas, notas, etc.

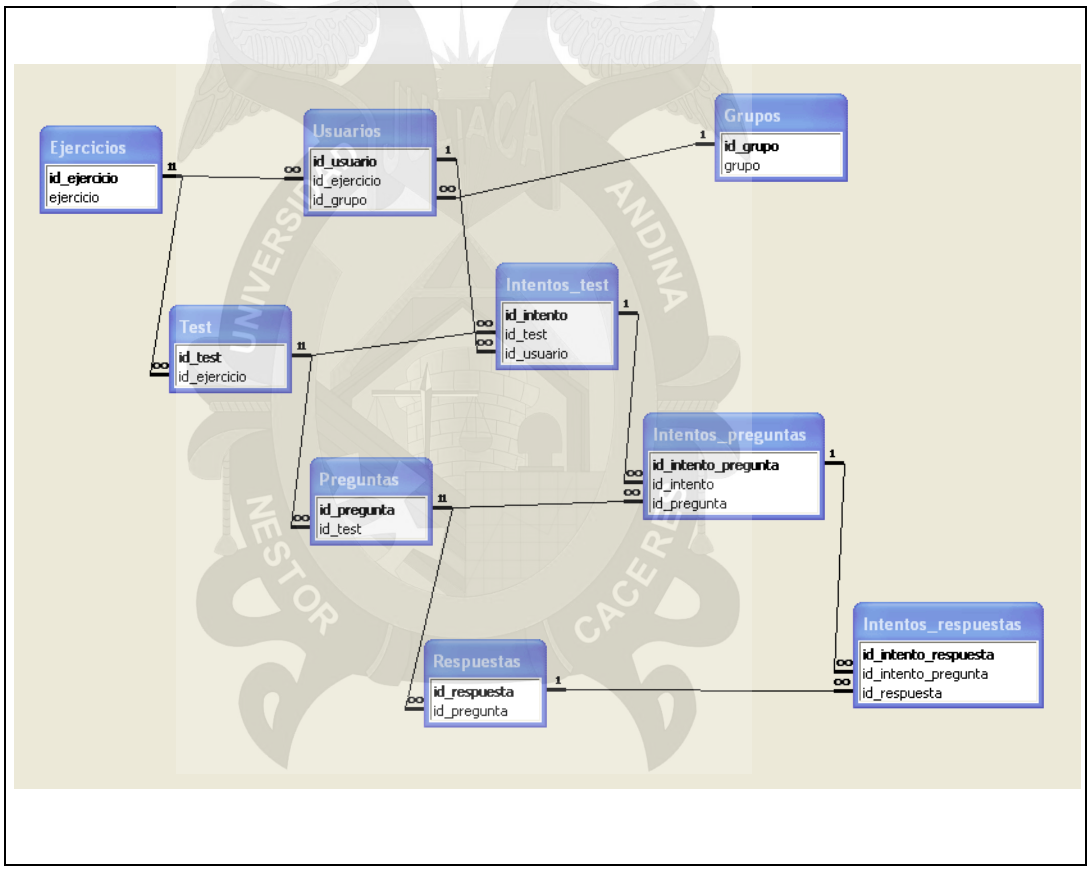
Se ha optado por hacer una única Base de Datos porque toda la información está relacionada y porque el volumen de información no es muy grande y cuando el WebMaster vaya a actualizarlo sólo deberá de actualizar una de ellas y no en varias, lo que supone un ahorro de tiempo.

4.1.1.- Base de Datos generación exámenes.

En la Base de Datos *generación exámenes* se va a tener la información de todos los usuarios como pueden ser sus nombres, apellidos, email, notas; de todos los test como pueden ser títulos, preguntas, respuestas, notas; y mucha más información referente a los intentos realizados por los alumnos, que posteriormente se necesitara para darlas a conocer a través de Internet.

Prácticamente en su totalidad, la información va a estar muy relacionada las unas con las otras, por ejemplo, un alumno matriculado en un ejercicio, un test dado de alta para un ejercicio, un alumno sólo podrá realizar test del ejercicio que este matriculado y así ocurre con toda la información, por lo que es preferente pasar a explicar el modelo entidad relación resultante del caso que se está realizando, para llegar a comprender mejor la estructura de la Base de Datos, mostrado en la **Fig. 4.1**.

Fig. 4.1.- Diagrama de la Base de Datos *generacionexamenes*



En el modelo entidad relación anterior se han colocado los atributos más importantes, debido al gran número de atributos que se tienen, los cuales, van a estar detallado en cada una de las tablas resultantes de este modelo.

A continuación, se van a explicar todas las tablas que tenemos del modelo entidad relación anterior, así como todos los campos de las mismas.

- **Tabla ejercicios**

En la tabla *ejercicios* se van a almacenar los distintos ejercicios académicos de los que se van a poder matricular los alumnos.

Los campos que se van a tener en esta tabla son:

- id_ejercicio: que va a ser un identificador único para cada ejercicio, por el cual, tendremos una referencia rápida al registro que ocupa un ejercicio en la tabla de los ejercicios.

- ejercicio: donde se guardará el valor del ejercicio académico.

- validado: nos india si el ejercicio esta validado o no. El desvalidar un ejercicio es una forma fácil de evitar la entrada a la web de todos los alumnos que estén dados de alta en ese ejercicio.

Fig. 4.2.- Tabla *ejercicios*

Campo	Tipo
id_ejercicio	int(10)
ejercicio	int(4)
validado	enum('N', 'S')

- **Tabla grupos**

En la tabla *grupos* se van a almacenar los distintos grupos en los que va a poder estar matriculados los alumnos.

Los campos que se van a tener en esta tabla son:

- id_grupo: que va a ser un identificador único para cada grupo, por el cual, tendremos una referencia rápida al registro que ocupa un grupo en la tabla de los grupos.

- grupo: donde se guardará el valor del grupo.

Fig. 4.3.- Tabla *grupos*

Campo	Tipo
id_grupo	int(10)
grupo	varchar(100)

- **Tabla usuarios**

En la tabla *usuario* se van a almacenar los distintos usuarios que van a poder entrar en la aplicación y de los cuales vamos a tener datos.

Los campos que se van a tener en esta tabla son:

- id_usuario: que va a ser un identificador único para cada usuario, por el cual, tendremos una referencia rápida al registro que ocupa un usuario en la tabla de los usuarios.

- login: donde se guardará el login del alumno que será un valor único, con respecto a todos los alumnos y a todos los administradores para que no haya problemas a la hora de hacer la búsqueda del usuario en cuestión. Este campo será asignado por el administrador mediante una carga de alumnos proveniente de un fichero Excel del tipo .csv

con un formato que hemos definido, en este caso, para el login se ha decidido que sea el DNI del usuario.

- pass: donde se guardará el password del alumno, que en un principio será asignado por el administrador mediante el proceso de carga anterior y posteriormente, podrá ser cambiado por los alumnos a su libre elección. Debiendo este tener cuidado de no revelárselo a nadie ya que si el password fuera cambiado por otro, el primero ya no podría volver a entrar como alumno con su password y carecería de los privilegios y servicios que antes poseía. Si esto diese lugar, lo que se debe hacer es comunicárselo a un profesor del Departamento para que este a su vez, se lo ponga de manifiesto al WebMaster del Nodo, problema al cual, pusiere solución en el menor tiempo posible. El password del usuario estará codificado mediante encriptación md4.

- email: donde se va a guardar el email de cada alumno, dato este que en la primera carga va a estar vacío y la aplicación va a obligar a rellenar al alumno con una dirección electrónica correcta la primera vez que entre en la aplicación.

- nombre: donde se va a guardar el nombre de cada usuario.

- apellidos: donde se va a guardar el apellido de cada usuario.

- administrador: donde se va a guardar si un usuario es administrador del web o es un simple alumno.

- id_grupo: es el identificador, por el cual, se conoce a que grupo pertenece un usuario.

- id_ejercicio: es el identificador, por el cual, se conoce el ejercicio al que puede acceder un usuario en la aplicación.

- validado: nos indica si un usuario puede entrar a la aplicación o no.

Fig.4.4.- Tabla *usuarios*

Campo	Tipo
id_usuario	int(10)
login	varchar(20)
pass	varchar(255)
email	varchar(100)
nombre	varchar(50)
apellidos	varchar(100)
administrador	enum('N', 'S')
id_grupo	int(10)
id_ejercicio	int(10)
validado	enum('N', 'S')

- **Tabla test**

En la tabla *test* se van a almacenar los distintos test que vamos a tener a disposición de los alumnos.

Los campos que se van a tener en esta tabla son:

- id_test: que va a ser un identificador único para cada test, por el cual, tendremos una referencia rápida al registro que ocupa un test en la tabla de los test.

- título: donde se va a guardar el título del test.

- id_ejercicio: es el identificador, por el cual, se conoce el ejercicio al que se debe computar el test en la aplicación.

- nivel: nos indica el nivel de complejidad que se le ha dado al test. El valor debe ser único por ejercicio y campo validado, de forma que para un mismo año, sólo debe de existir un único test con nivel de complejidad por ejemplo 1.

- max_intentos: nos indica el número máximo de intentos del que van a disponer los usuarios para realizar el test, siempre y cuando no haya superado el test. Si un usuario alcanzará este valor, se le comunica al alumno y al administrador de tal situación, de forma, que hasta que el administrador no le asigne un intento extra a dicho usuario, esté no podrá seguir realizando test.

- min_aprobado: nos indica la nota mínima desde donde se considera aprobado un test. El valor debe ser mayor que 0 y menor o igual a 10.

- num_preguntas: nos indica el número de preguntas necesarias para componer un test, es decir, la aplicación va a generar de forma dinámica a petición de los alumnos, test de este número de preguntas.

- grado_penalizacion: nos indica el tanto por ciento de penalización que se le dará a las respuestas incorrectas, pudiendo oscilar entre 0 y 100.

- duracion: nos indica la duración en minutos del test, es decir, el alumno dispondrá para realizar el test de está duración, si el alumno

no finaliza el test antes de que se agote este tiempo, la aplicación lo hará por el alumno.

- validado: nos indica si un test se puede realizar o no.

Toda la información de esta tabla es administrada por los usuarios del tipo administrador, pudiendo cambiar la información a su antojo de cada test, siempre y cuando, esté no haya sido realizado por algún usuario, ya que, creo que es vital, que no se pueda modificar la información, para que esta sea lo más veraz posible y no se pueda falsear los resultados de los alumnos.

Fig. 4.5.- Tabla *test*

Campo	Tipo
id_test	int(10)
titulo	varchar(250)
id_ejercicio	int(10)
nivel	int(10)
max_intentos	int(10)
min_aprobado	int(10)
num_preguntas	int(10)
grado_penalizacion	int(11)
duracion	int(10)
validado	enum('S', 'N')

- **Tabla preguntas**

En la tabla *preguntas* se van a almacenar las distintas preguntas asociadas a cada uno de los test que vamos a tener a disposición de los alumnos.

Los campos que se van a tener en esta tabla son:

- id_pregunta: que va a ser un identificador único para cada pregunta, por el cual, tendremos una referencia rápida al registro que ocupa una pregunta en la tabla de las preguntas.

- id_test: es el identificador, por el cual, se asocia la pregunta al test.

- pregunta: donde se va a guardar la pregunta que se le hace al alumno.

- tipo: donde se va a guardar el tipo de la pregunta. El tipo de la pregunta puede ser Simple (S), que quiere decir que sólo tiene una respuesta correcta o puede ser Compuesta (C), que quiere decir que puede tener más de una repuesta correcta.

Fig. 4.6.- Tabla *preguntas*

Campo	Tipo
id_pregunta	int(10)
id_test	int(10)
pregunta	text
tipo	enum('S', 'C')

- **Tabla respuestas**

En la tabla *respuestas* se van a almacenar las distintas respuestas de las preguntas asociadas a cada uno de los test que vamos a tener a disposición de los alumnos.

Los campos que se van a tener en esta tabla son:

- id_respuesta: que va a ser un identificador único para cada respuesta, por el cual, tendremos una referencia rápida al registro que ocupa una respuesta en la tabla de las respuestas.

- id_pregunta: es el identificador, por el cual, se asocia la respuesta a la pregunta.

- respuesta: donde se va a guardar las respuestas que tiene la pregunta.

- correcta: donde se va a guardar si la respuesta es correcta o no. Con este dato podremos saber si el alumno ha acertado o no a la hora de realizar el test; y poder así, obtener la nota final de este.

Fig. 4.7.- Tabla *respuestas*

Campo	Tipo
id_respuesta	int(10)
id_pregunta	int(10)
respuesta	text
correcta	enum('S', 'N')

- **Tabla intentos_test**

En la tabla *intentos_test* se van a almacenar los resultados de los distintos intentos que han realizado los alumnos para superar los test.

Los campos que se van a tener en esta tabla son:

- id_intento: que va a ser un identificador único para cada intento, por el cual, tendremos una referencia rápida al registro que ocupa un intento de un test en la tabla de los intentos_test.

- id_test: es el identificador, por el cual, se asocia el intento a un test.

- id_usuario: es el identificador, por el cual, se asocia el intento a un usuario.

- f_comienzo: donde se va a guardar la fecha de comienzo del intento del test.

- f_fin: donde se va a guardar la fecha de fin del intento del test.

- nota: donde se va a guardar la calificación obtenida por el alumno en el test realizado.

- intento: donde se va a guardar el número de intento que lleva el alumno para intentar superar dicho test. Este campo si vale 0, nos va indicar que para dicho test y alumno se han necesitado de intentos extras de los que originariamente el administrador había asignado.

- evaluado: donde se va a guardar si el test ha sido evaluado o no. Siempre debe estar a evaluado, de lo contrario es que ha sucedido algo, como que el alumno haya cortado la conexión con la aplicación antes de corregir el test, se haya ido la luz, etc.

- aprobado: donde se va a guardar si el test ha sido superado por el alumno o por si el contrario debe volver a intentarlo.

Fig. 4.8.- Tabla *intentos_test*

Campo	Tipo
id_intento	int(10)
id_test	int(10)
id_usuario	int(10)
f_comienzo	datetime
f_fin	datetime
nota	float
intento	int(10)
evaluado	enum('S', 'N')
aprobado	enum('S', 'N')

- **Tabla intentos preguntas**

En la tabla *intentos preguntas* se van a almacenar los resultados que concierne a las preguntas de los distintos intentos que han realizado los alumnos para superar los test.

Los campos que se van a tener en esta tabla son:

- id_intento_pregunta: que va a ser un identificador único para cada pregunta de cada intento, por el cual, tendremos una referencia rápida al registro que ocupa una pregunta de un test en la tabla de los intentos preguntas.

- id_intento: es el identificador, por el cual, se asocia la pregunta a un intento de test.

- id_pregunta: es el identificador, por el cual, se asocia el intento pregunta a una pregunta.

- puntuacion: donde se va a guardar la calificación obtenida por el alumno en dicha pregunta.

Fig. 4.9.- Tabla *intentos_preguntas*

Campo	Tipo
id_intento_pregunta	int(10)
id_intento	int(10)
id_pregunta	int(10)
puntuacion	float

- **Tabla intentos respuestas**

En la tabla *intentos_respuestas* se van a almacenar los resultados que concierne a las respuestas de las preguntas de los distintos intentos que han realizado los alumnos para superar los test.

Los campos que se van a tener en esta tabla son:

- id_intento_respuesta: que va a ser un identificador único para cada respuesta de la pregunta de cada intento, por el cual, tendremos una referencia rápida al registro que ocupa una respuesta de una pregunta de un test en la tabla de los intentos_respuestas.

- id_intento_pregunta: es el identificador, por el cual, se asocia la respuesta a una pregunta de un intento de test.

- id_respuesta: es el identificador, por el cual, se asocia el intento respuesta a una respuesta.

- acertada: donde se va a guardar si el alumno ha acertado o no la respuesta.

- puntuacion: donde se va a guardar la calificación obtenida por el alumno en dicha respuesta.

Fig. 4.10.- Tabla *intentos_respuestas*

Campo	Tipo
id_intento_respuesta	int(10)
id_intento_pregunta	int(10)
id_respuesta	int(10)
acertada	enum('S', 'N')
puntuacion	float

4.2.- Análisis del Nodo Web.

El análisis del Nodo Web del Departamento de Matemática Aplicada I se va a dividir en dos grandes ramas:

- Estructura del Nodo Web, donde se explicará la jerarquía del Web para hacer comprender mejor como está compuesto internamente el Web.

- Funcionalidad del Nodo, donde se explicará con el mayor lujo de detalles todo lo que hace el Web, de manera, que se le pueda sacar el máximo provecho a nuestra aplicación.

4.2.1.- Estructura del Nodo Web.

El Nodo Web del Departamento de Matemática Aplicada I consta de una estructura jerárquica dividida claramente en dos partes, la pública que va a tener acceso todos los alumnos que se hayan logado y la privada que es sólo para los administradores de la aplicación, desde donde se podrá gestionar toda la información necesaria para el correcto funcionamiento de la aplicación.

A continuación, se pasa a explicar todas las páginas del Nodo Web, una a una, empezando por la primera, que es un tanto especial:

- **Login**

Donde los usuarios van a tener que identificarse como usuarios de la aplicación si quieren poder acceder a los servicios que se presta en la misma. Al final si el usuario es correctamente validado se le lleva a la página de inicio (*default.php*), donde según si es un alumno o un administrador, podrá hacer tener acceso a una serie de privilegios.

- **Inicio**

Donde se va a tener acceso por parte de los alumnos a diferentes enlaces, los cuales, le van a permitir realizar los test, cambiar sus datos y ver las notas de los test realizados anteriormente.

- **Mis Datos**

Donde cada usuario va a tener acceso a conocer los datos personales que se dispone de sí mismo y cambiar aquellos que no estén rellenos o que sean incorrectos.

- **Test**

Donde los alumnos van a tener acceso a realizar los test que los administradores han dado de alta en la aplicación según el ejercicio en que estén, ya que, según el ejercicio, podrá realizar unos test u otros.

- **Notas**

Donde los alumnos van a tener acceso a ver las notas de todos los test que han realizado y llevar un seguimiento de sus exámenes.

- **Administración**

Donde sólo los administradores van a tener acceso y van a poder llevar a cabo todas las acciones de mantenimiento de los datos de la aplicación, como llevar a cabo el alta, edición y eliminación de grupos de trabajo, ejercicios, usuarios, test, preguntas y respuestas.

4.2.2.- Funcionalidad del Nodo Web.

La funcionalidad del Nodo Web se va a ir explicando página a página, según el orden lógico de actuación de los administradores y alumnos y una vez explicado cada página se comentará las contenidas en ellas, ya que, son dependientes y tienen mucho que ver en gran medida con ellas.

- **Login**

La función de esta página será la de permitir el acceso a los usuarios de la aplicación y restringir el acceso a aquellos que no disponga de permisos. Cuando un usuario es validado rellenamos unas variables de

sesión que tendremos acceso a ellas en todas las páginas siguientes y que nos va a dar una información primordial, como, por ejemplo, si el usuario es administrador o no.

- **Default**

La función de la página Default es la de introducirnos un poco más en la aplicación, dando al usuario la bienvenida y mostrándole las distintas acciones que puede realizar según el perfil del mismo, estás acciones van a estar como plantilla en todas las páginas de la aplicación de forma que el usuario desde todas las páginas pueda acceder a todos los enlaces.

También se dispone de un contador de visitas, el cual está en la parte inferior izquierda de la aplicación y que se incrementa cada vez que un usuario accede correctamente a la aplicación.

- **Mis Datos**

Las funciones de esta página van a ser la de mostrar a los usuarios sus propios datos personales, como el nombre, apellidos, correo electrónico, etc., y la posibilidad de cambiarlos cuando sean incorrectos.

La primera vez que un usuario no administrador entra a la aplicación, por defecto se le va a llevar a esta página para que rellene la totalidad de sus datos y pueda cambiar la contraseña y hasta que no lo haga no lo dejaremos accede a la aplicación.

Al final se nos muestra un botón de Guardar, el cual, al pulsarlo será el encargado de validar la información introducida y si todo está correcto, entonces se grabarán los datos del alumno.

- **Test**

Las funciones de esta página van a ser la de mostrar a los alumnos la información del test siguiente que puede realizar si es que existe algún test para ese propósito. Si existe test que pueda generar, se le activará un botón de generar test, el cual al pulsarlo le generará al alumno un test de forma dinámica con el número de preguntas que el administrador ha definido y que debe resolver en una duración especificada por el administrador.

Una vez generado el test la aplicación nos carga el test generado en otra página, donde un contador va disminuyendo hasta llegar a cero y es en este momento donde se corrige el test de forma automática. En esta página el usuario puede tener dos tipos de preguntas, las simples y las compuestas, las primeras son aquellas que sólo van a disponer de una respuesta correcta y por tanto sólo dejaremos a los alumnos seleccionar una respuesta y las segundas que van a poder tener más de una respuesta correcta y por tanto dejaremos que seleccionen todas las respuestas que consideren correctas.

Una vez que hayan contestado a todas las preguntas es cuando el alumno debe proceder a pulsar el botón Corregir, el cual, corregirá el examen y pasará el controla la página de Notas, para que el alumno pueda ver su resultado.

- **Notas**

Las funciones de esta página van a ser la de mostrar a los alumnos los resultados de los distintos intentos que han llevado a cabo para superar los distintos test.

Aquí además de saber si hemos superado o no los test, podremos ver información adicional de los distintos intentos que hemos realizados, como

pueden ser la fecha de inicio, fecha final, nota, si se ha aprobado o no el test, el número de intento que se lleva y si se han necesitados un mayor número de intentos de los que el profesor había asignado.

- **Administración**

A partir de ahora las páginas que voy a comentar son del grupo de Administración. Las cuales van a disponer de un mecanismo de seguridad que van a impedir el acceso a usuarios que no sean administradores.

Una vez que entramos en el menú de Administración mediante el botón de administración que aparece en la botonera de la parte superior, nos muestra un listado de acciones que se van a poder realizar, las cuales mostramos en detalle con los siguientes bloques.

- **Gestión de Usuarios**

Las funciones de este grupo de páginas van a ser la de gestionar toda la información de los usuarios, el administrador va a poder dar de alta de forma manual a alumnos, va a poder eliminar alumnos que ya no sirvan para el sistema, podrá listar los alumnos según algún criterio de búsqueda, etc.

Una vez que hayamos listados los usuarios, vamos a poder ordenarlo por los campos de la tabla, con tan sólo pinchar sobre el título de la columna que queramos ordenar, además, si los registros no caben en una sola página vamos a poder paginar y también vamos a poder modificar el número de registros que se muestran por página.

Para editar un usuario, sólo basta localizarlo en la página y luego pinchando sobre cualquier dato de este, nos va a llevar a una página donde vamos a poder editar los campos del usuario, además vamos a tener un enlace para poder ver todas las notas del usuario seleccionado.

- **Carga de Fichero de Usuarios**

Esta página va a dar la posibilidad al administrador de carga una lista de usuarios provenientes de un fichero en la aplicación en un grupo, ejercicio y con un valor en el campo validado. Es una forma rápida de dar de alta a todos los usuarios de una asignatura.

El fichero deberá venir en un fichero tipo Excel con extensión *csv*, donde el primer campo será el DNI, el cual utilizaremos como Login y Contraseña y el segundo campo serán los apellidos y el nombre separados por coma, los cuales separaremos en el proceso de carga y nos servirá como apellidos y nombre respectivamente.

Para cargar un fichero debemos seleccionar el fichero que cumple con las condiciones anteriores, pinchando sobre el botón *Examinar* y eligiendo el fichero en la ruta donde se encuentre, luego debemos seleccionar el grupo, el ejercicio y si estarán o no validados los usuarios que vamos a cargar. Posteriormente, pulsamos sobre el botón *Cargar*, el cual, cargará los usuarios en el sistema y nos mostrará lo que ha ocurrido, si todos han sido cargados o si algunos ya estaban en el sistema y no han sido posible su carga.

- **Gestión de Ejercicios**

Las funciones de este grupo de páginas van a ser la de gestionar toda la información de los ejercicios, el administrador va a poder dar de alta los ejercicios, va a poder eliminar ejercicios que ya no sirvan para el sistema, podrá listar los ejercicios, etc.

Una vez que hayamos listados los ejercicios, vamos a poder ordenarlo por los campos de la tabla, con tan sólo pinchar sobre el título de la columna que queramos ordenar, además, si los registros no caben en una

sola página vamos a poder paginar y también vamos a poder modificar el número de registros que se muestran por página.

Para editar un ejercicio, sólo basta localizarlo en la página y luego pinchando sobre cualquier dato de este, nos va a llevar a una página donde vamos a poder editar los campos del ejercicio.

- **Gestión de Grupos**

Las funciones de este grupo de páginas van a ser la de gestionar toda la información de los grupos, el administrador va a poder dar de alta los grupos, va a poder eliminar grupos que ya no sirvan para el sistema, podrá listar los grupos, etc.

Una vez que hayamos listados los grupos, vamos a poder ordenarlo por los campos de la tabla, con tan sólo pinchar sobre el título de la columna que queramos ordenar, además, si los registros no caben en una sola página vamos a poder paginar y también vamos a poder modificar el número de registros que se muestran por página.

Para editar un grupo, sólo basta localizarlo en la página y luego pinchando sobre cualquier dato de este, nos va a llevar a una página donde vamos a poder editar los campos del grupo.

- **Gestión de Test**

Las funciones de este grupo de páginas van a ser la de gestionar toda la información de los test, el administrador va a poder dar de alta los test, va a poder eliminar test que ya no sirvan para el sistema, podrá listar los test según algún criterio de búsqueda, etc.

Una vez que hayamos listados los test, vamos a poder ordenarlo por los campos de la tabla, con tan sólo pinchar sobre el título de la columna que queramos ordenar, además, si los registros no caben en una sola página vamos a poder paginar y también vamos a poder modificar el número de registros que se muestran por página.

Para editar un test, sólo basta localizarlo en la página y luego pinchando sobre cualquier dato de este, nos va a llevar a una página donde vamos a poder editar los campos del test.

Además, en la página donde podremos editar los test, vamos a tener un enlace al detalle del test, donde podremos dar de alta o modificar las preguntas y las respuestas que componen el test y un enlace a las notas, donde se podrán ver todas las notas de todos los alumnos que han realizado dicho test.

Nota: Si eres un usuario registrado es muy importante que antes de abandonar el Nodo Web, cierres tú Sesión de Usuario, para evitar que terceras personas puedan acceder a sus datos. Para cerrar su Sesión de Usuario sólo debes pinchar en el botón *Cerrar sesión*, que se encuentra en todas las páginas en la parte derecha de la botonera.

CAPITULO V IMPLEMENTACION DEL SISTEMA

5.- DESCRIPCION DEL SISTEMA.

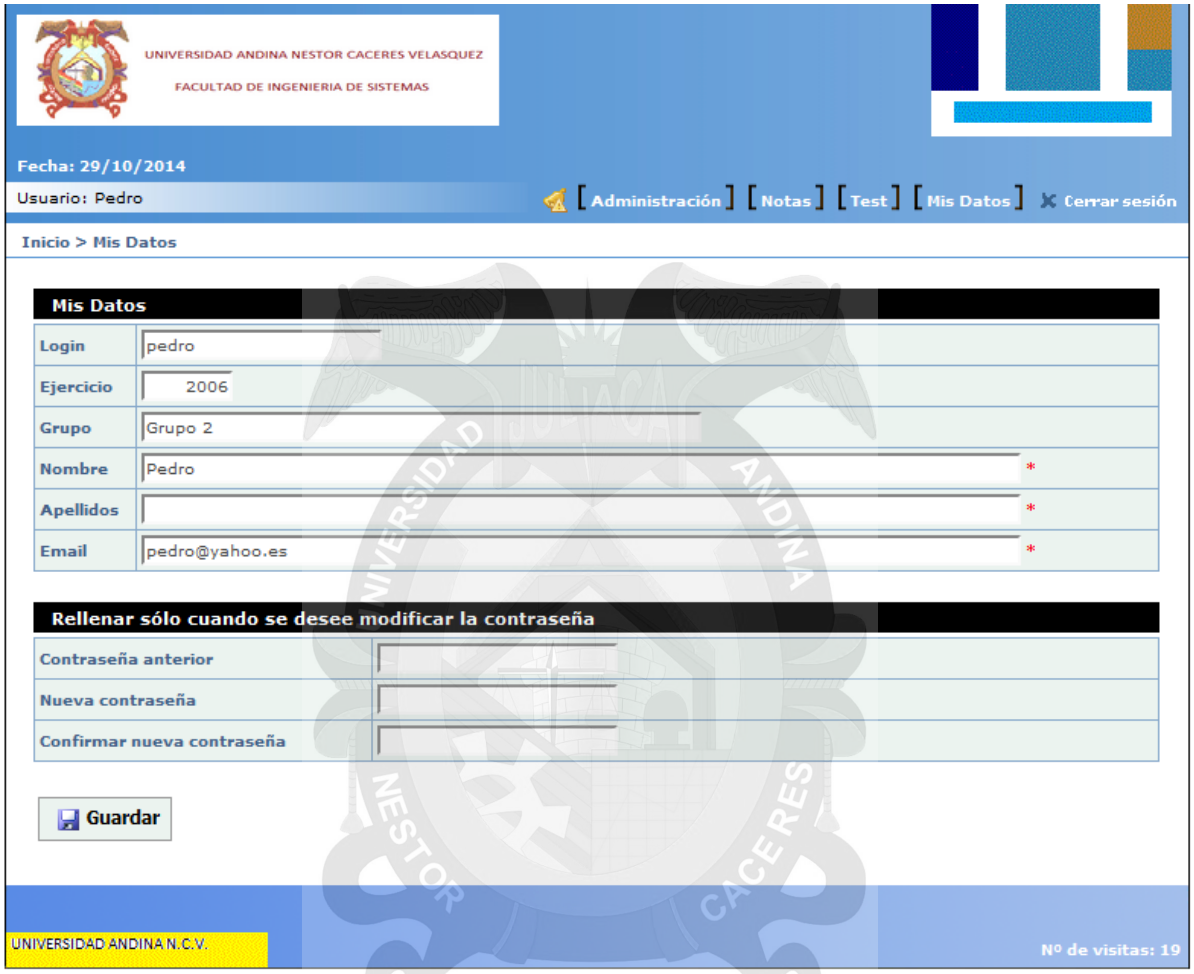
La aplicación tiene claramente separadas dos partes, la del alumno encargado de realizar los test y la del administrador, quien tendrá que mantener toda la información del sistema En ambos casos se accederá a través de la misma pantalla, ver figura *Fig 5.1*. En ella aparecerá un formulario de identificación donde tendrán que rellenar su nombre de usuario y contraseña. Dependiendo de los datos introducidos, se le activará la opción de *Administración* o no.

Fig. 5.1.- Página de acceso a la aplicación

Una vez que el usuario es validado por la aplicación y el sistema comprueba que no es un administrador, el sistema manda al usuario a la página de sus datos personales, si el sistema detecta que el alumno no tiene relleno alguno de sus datos, además, en este momento el alumno puede optar por cambiar su contraseña y los datos que observe que no son correctos (ver figura *Fig 5.2*).

Como se puede observar en la figura *Fig 5.2*, las únicas acciones que el usuario puede realizar son la de cambiar los datos o cerrar la aplicación.

Fig. 5.2.- Página de datos personales de un usuario



Una vez que el usuario ha rellenado todos sus datos y ha procedido a guárdalos, el sistema le lleva a la página de inicio, desde donde se le dará la bienvenida por parte del Departamento de Matemática Aplicada I y tendrá accesible todas las acciones que el usuario puede llevar a cabo (ver figura *Fig 5.3*).

Como se puede observar el usuario puede cerrar la sesión, consultar sus datos, que no es más que navegar a la página mostrada en la Figura *Fig 5.2*,

realizar los Test y ver las notas. Además, se puede observar una serie de iconos, que tal como comenté anteriormente nos van a llevar a la página de inicio, según donde pinchemos.

También se puede observar en la esquina inferior derecha un contador, el cual nos indica el número de visitas que ha recibido la aplicación.

Fig. 5.3.- Página de inicio

Para que el alumno pueda realizar un test, debe pinchar sobre el botón *Test* de la botonera, el cual le llevará a la pantalla de la figura *Fig 5.4*, donde la aplicación informará al alumno de cuál es el siguiente Test que tiene que realizar, si es que fuese el caso, porque puede pasar que no se hayan definido test (ver figura *Fig 5.4*) o que el alumno haya superado con éxito todos los test que tiene pendiente (ver figura *Fig 5.4*) o que se hayan agotado el número de intentos de realización de un Test y se deba de esperar que el administrador proporcionen intentos extras para este alumno y este test (ver figura *Fig 5.6*).

Fig. 5.4.- Página donde se informa al alumno del siguiente test que puede realizar

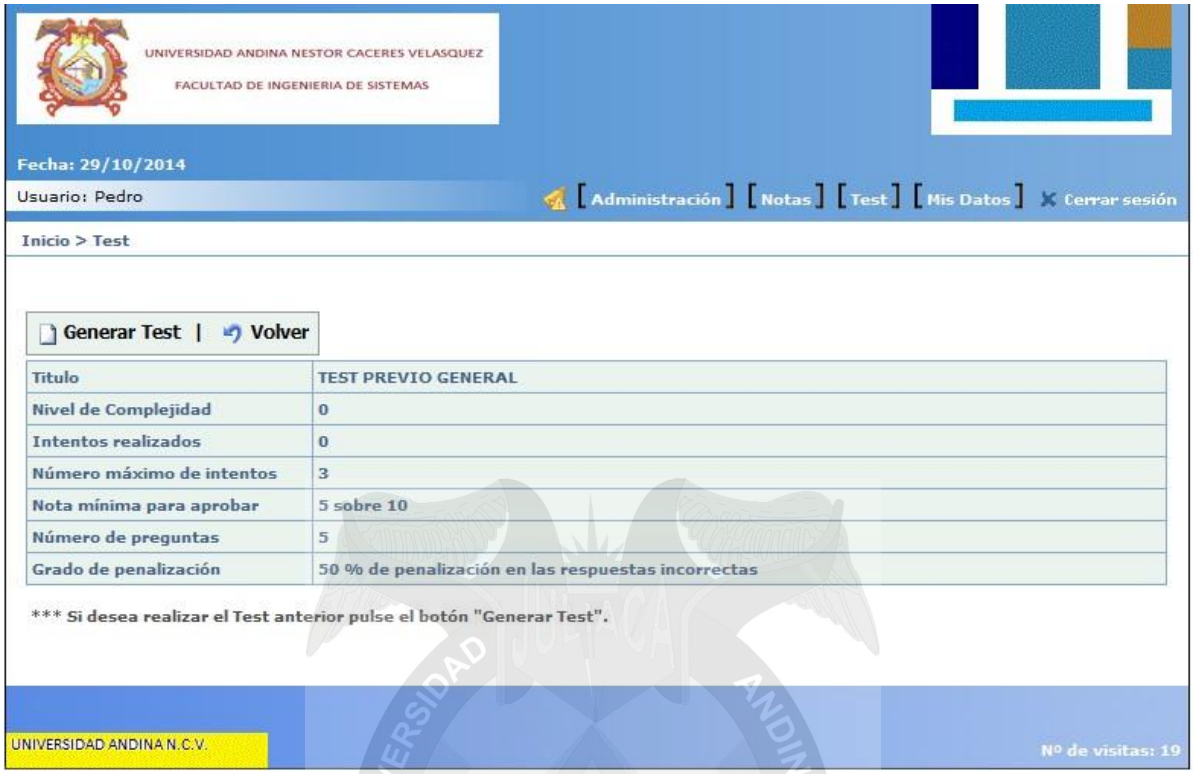


Fig. 5.5.- Página donde se informa al alumno que no se ha encontrado test que puede realizar

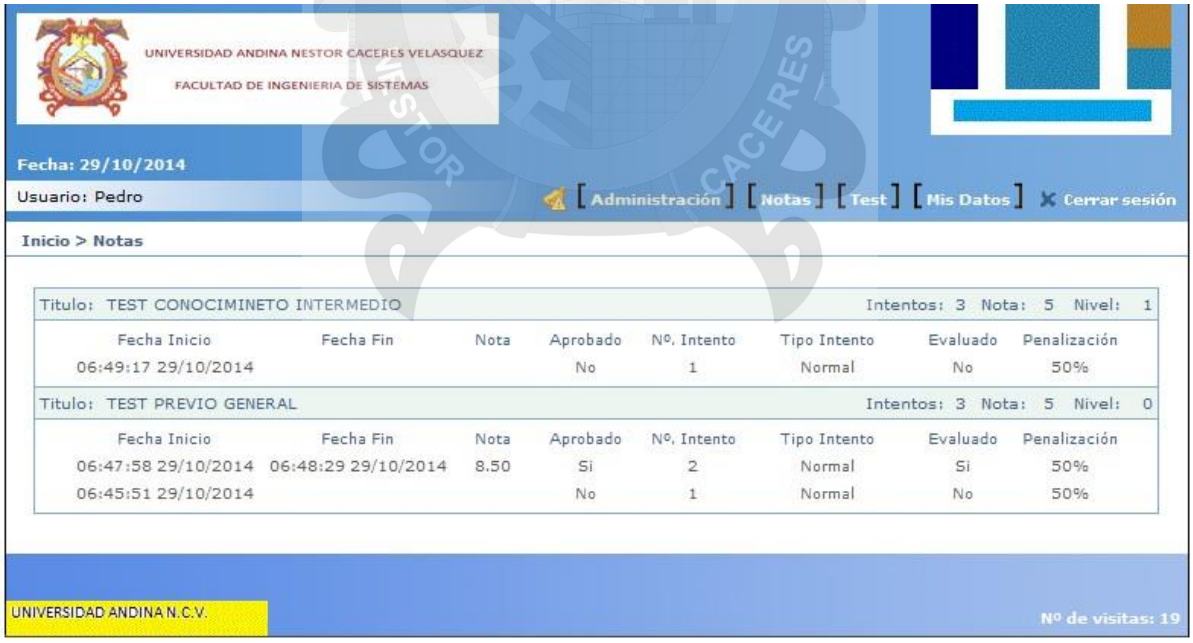
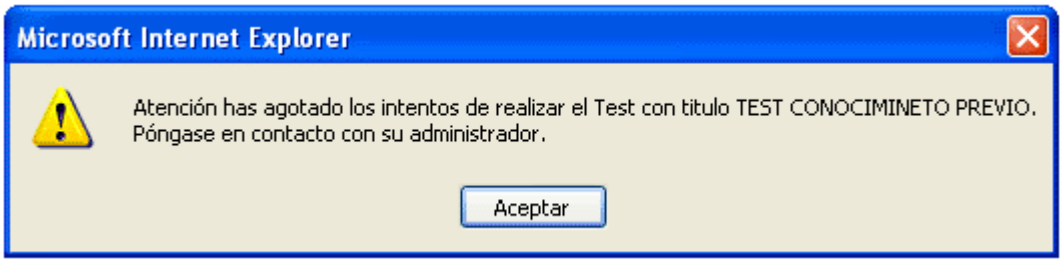
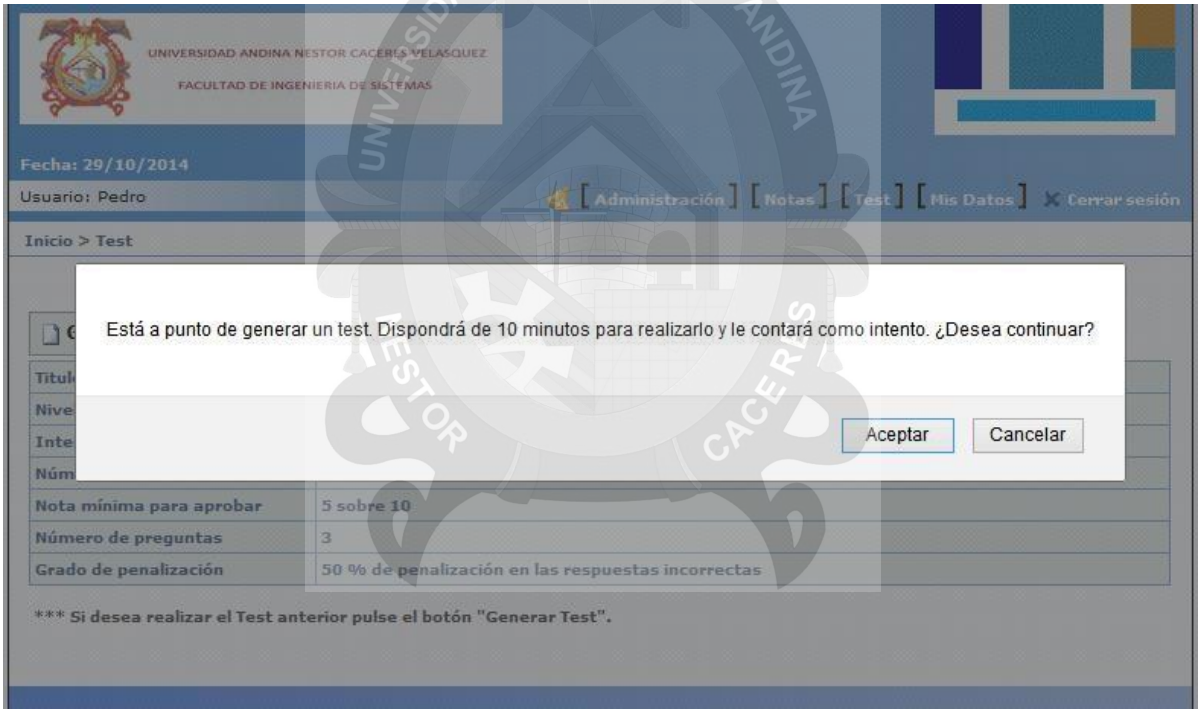


Fig. 5.6.- Página donde se informa al alumno que ha agotado el número de intento para realizar un test



Si se ha encontrado test y queremos realizarlo, tenemos que pulsar sobre el botón *Generar test* de la página de la figura *Fig 5.4*, el cual nos preguntará si estamos seguro de querer generar un test y nos informará del tiempo que dispondremos para realizarlo.

Fig. 5.7.- Página donde se informa al alumno de la duración del test y si esta seguro de querer hacerlo



La figura *Fig 5.8* muestra un test generado dinámicamente, donde el alumno debe de contestar las preguntas antes de que se le agote el tiempo y luego pulsar el botón Corregir, el cual, le llevará a la pantalla de la Figura Fig 5.9, donde podrá ver las notas de todos los test que ha realizado.

La pantalla donde el alumno puede ver las notas, también es accesible por el botón Notas de la botonera superior que existe en todas las páginas de la aplicación y por el hipervínculo que existe en la pantalla de inicio.

En la pantalla de la figura *Fig 5.8*, podemos observar los distintos iconos que están asociados a las respuestas, estos iconos nos van a indicar si se trata de una pregunta simple o por el contrario se trata de una pregunta compuesta, es decir una o varias respuestas son correctas.

Si no queremos utilizar las preguntas del tipo simple, para evitar que el alumno sepa que sólo tiene una respuesta correcta, podemos declarar la pregunta como compuesta y definir una única respuesta correcta.

Con esto damos por finalizado el manual de usuario.

Fig. 5.8.- Página donde se muestra un test generado de forma dinámica

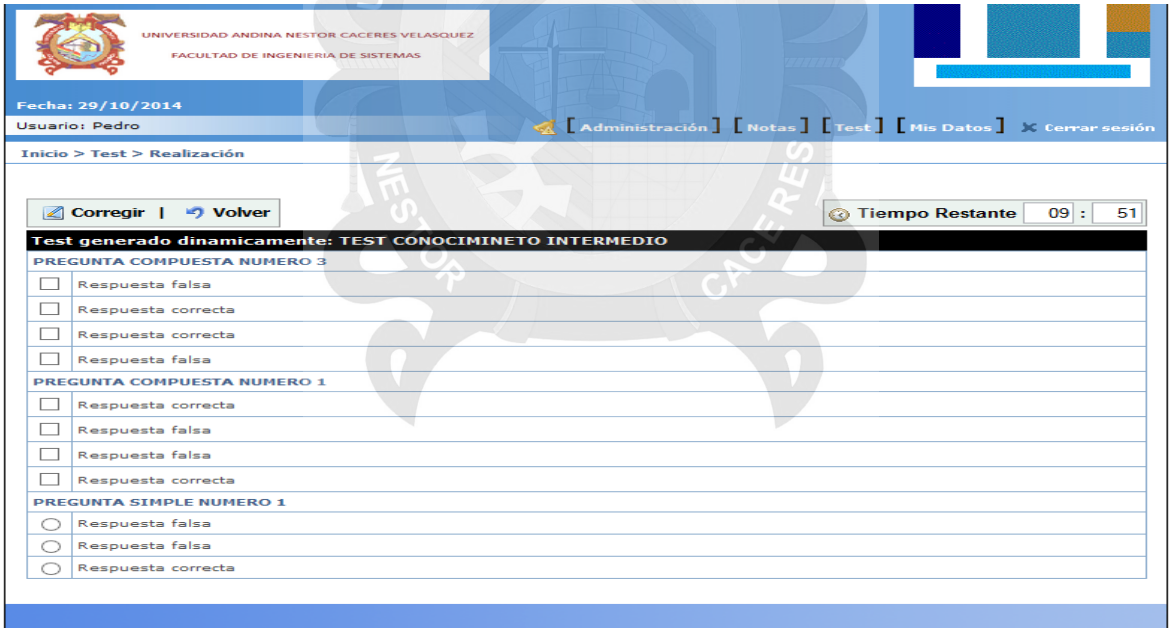
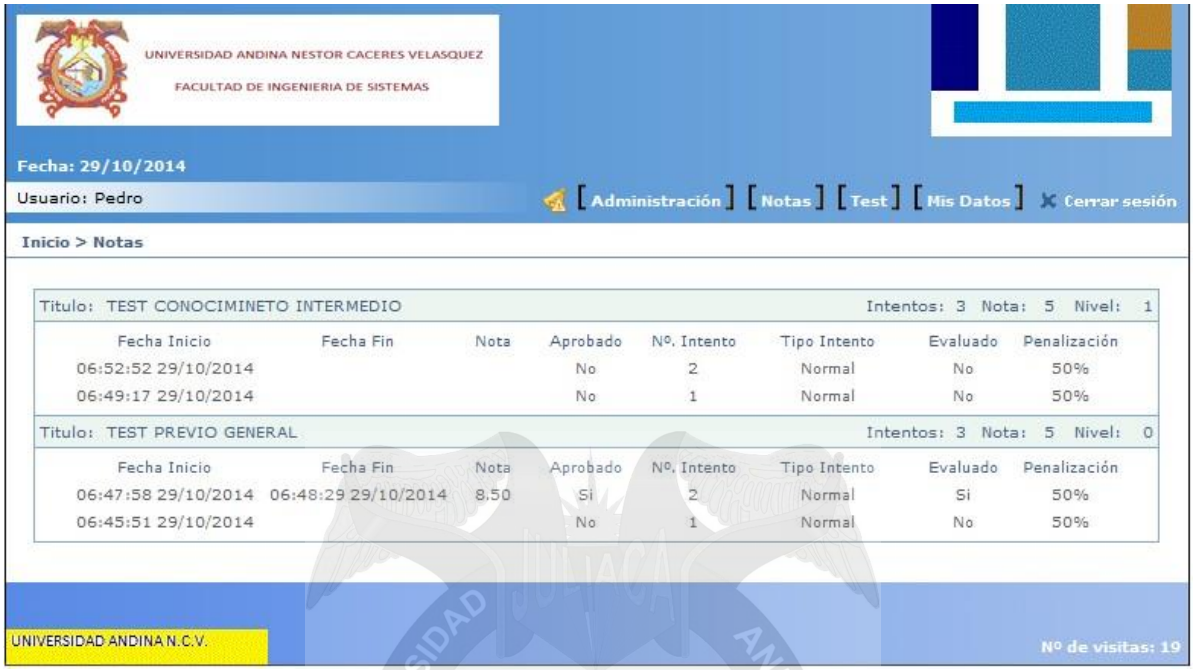


Fig. 5.9.- Página donde se muestra las notas de un alumno



Una vez que administrador haya accedido a la aplicación por la página de la figura *Fig 5.1*, el sistema le llevará a la página de inicio, donde este podrá disponer de todos los servicios que tenía un alumno además de otros por ser administrador.

Fig. 5.10.- Página de inicio de un administrador

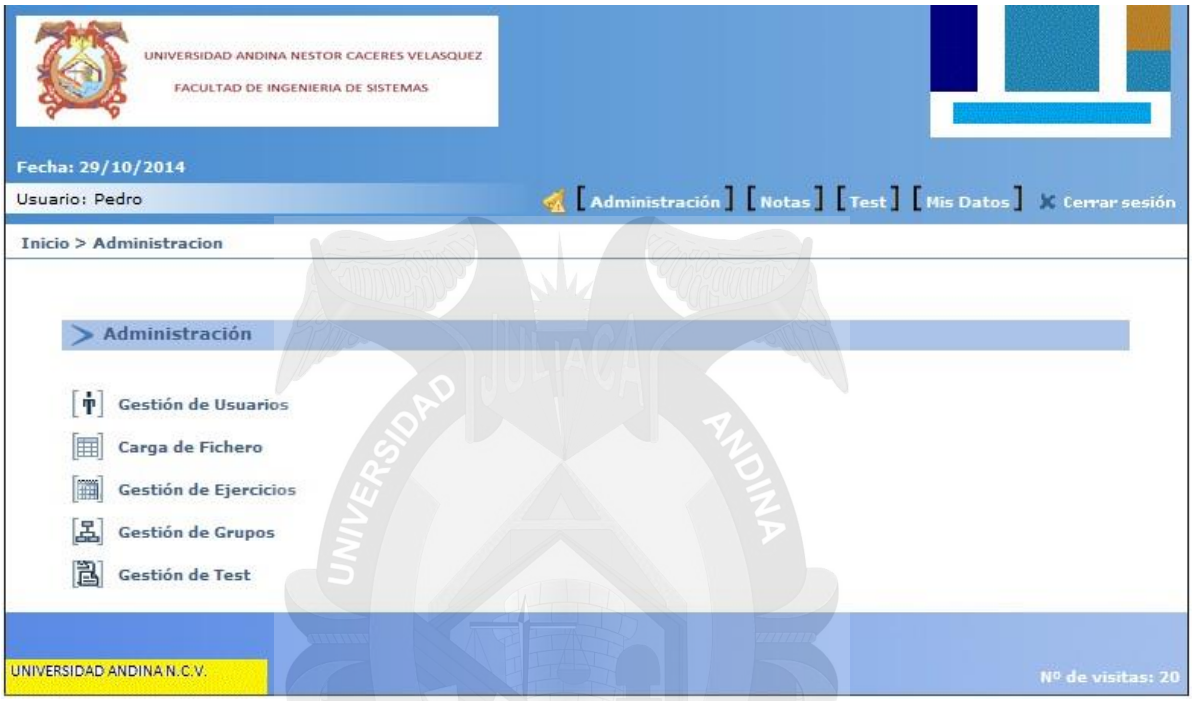
En la página de la figura anterior se puede observar como tiene algunos elementos más que un alumno, el icono de la campana, nos lleva a una página emergente, la cual, al entrar en la aplicación se abre, si es que tiene datos que mostrar y donde el administrador puede ver quiénes son los alumnos que han agotado el número de intentos para realizar un test sin haberlo superado (ver figura *Fig 5.11*). Además, podemos ver un icono de administración donde el administrador podrá llevar a cabo todo el mantenimiento de los datos del Nodo Web (ver figura *Fig 5.12*).

La página de la figura *Fig 5.11*, nos muestra una serie de alertas, las cuales, le indica al administrador los usuarios y para que test se han quedado sin intentos. Luego, el administrador si ve conveniente darle al usuario un intento más para realizar el test, sólo debe de pinchar sobre el botón *Añadir*.

Fig. 5.11.- Página de alertas de intentos agotados

La página de la figura *Fig 5.12*, nos muestra una serie de opciones que el administrador va a tener y sin las cuales, no podría administrar el Nodo Web.

Fig. 5.12.- Página donde se lista las acciones que puede llevar a cabo un administrador



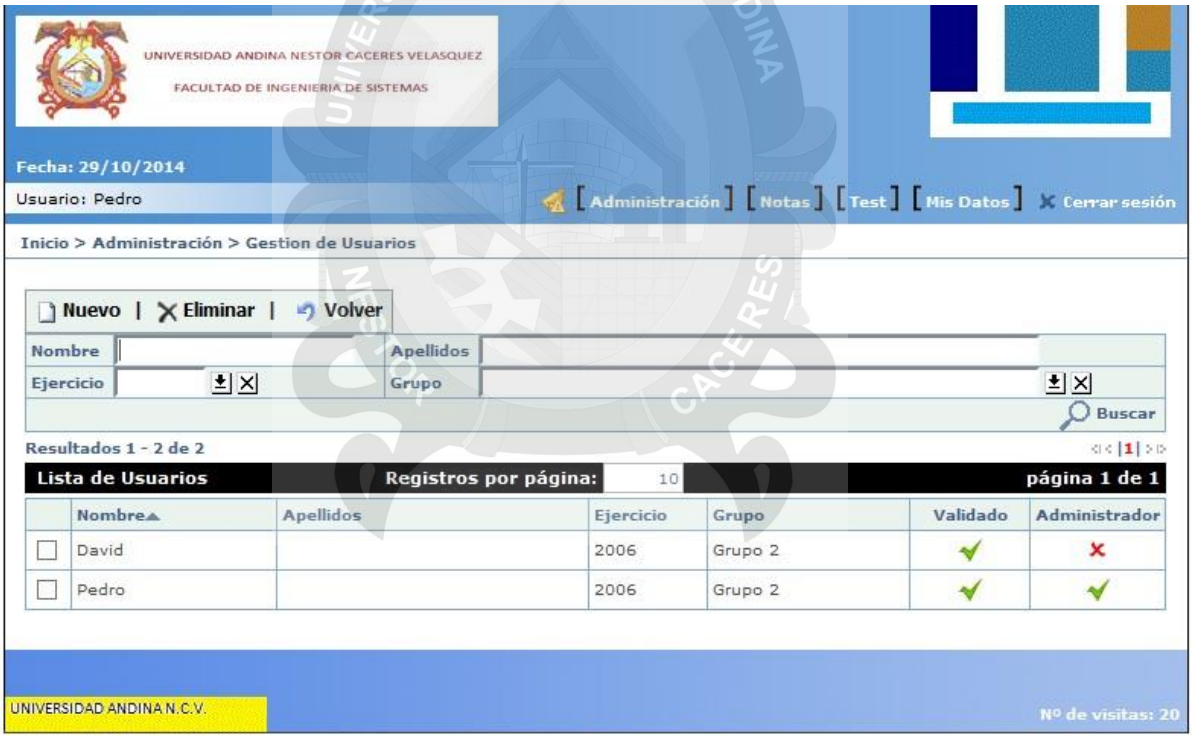
En la página de la figura *Fig 5.13* podemos ver la página donde se gestiona los usuarios, en ella podemos buscar por nombre, apellidos, ejercicio o grupo al que pertenece el usuario, para ello sólo debemos de rellenar los campos por los que queremos realizar la búsqueda o seleccionar de una lista el ejercicio o grupo y pulsar el botón *Buscar*.

Cuando entremos en la página o cuando realicemos una búsqueda nos mostrara un listado con todos los usuarios y podremos ir navegando por las diferentes páginas para ver el resto de usuarios, incluso podremos cambiar la ordenación o el número de registros que se muestra en cada pantalla.

El administrador va a poder eliminar un usuario con sólo seleccionarlo mediante el check de la izquierda y luego pinchando sobre el botón *Eliminar*, siempre y cuando el sistema no encuentre dependencias con el registro que queremos borrar, por ejemplo, para borrar un usuario primero debemos borrar todas sus notas.

Para editar un usuario basta con pinchar sobre cualquier campo del usuario en cuestión, lo veremos detallado en la figura *Fig 5.15* y para dar un usuario de alta de forma manual, tendremos que pinchar sobre el botón *Nuevo* que nos llevará a la página de la figura *Fig 5.15* pero con los campos en blanco y sin la opción de ver las Notas, ya que aún no existe el usuario.

Fig. 5.13.- Página donde se gestiona los usuarios



En la página de la figura *Fig 5.14* se puede observar la página que resulta al pulsar sobre el botón de lista de ejercicios de la página de la figura anterior. En dicha página vamos a poder cambiar la ordenación y para

seleccionar un ejercicio sólo debemos pinchar en la fila del ejercicio que queremos coger.

Fig. 5.14.- Página donde se selecciona un ejercicio

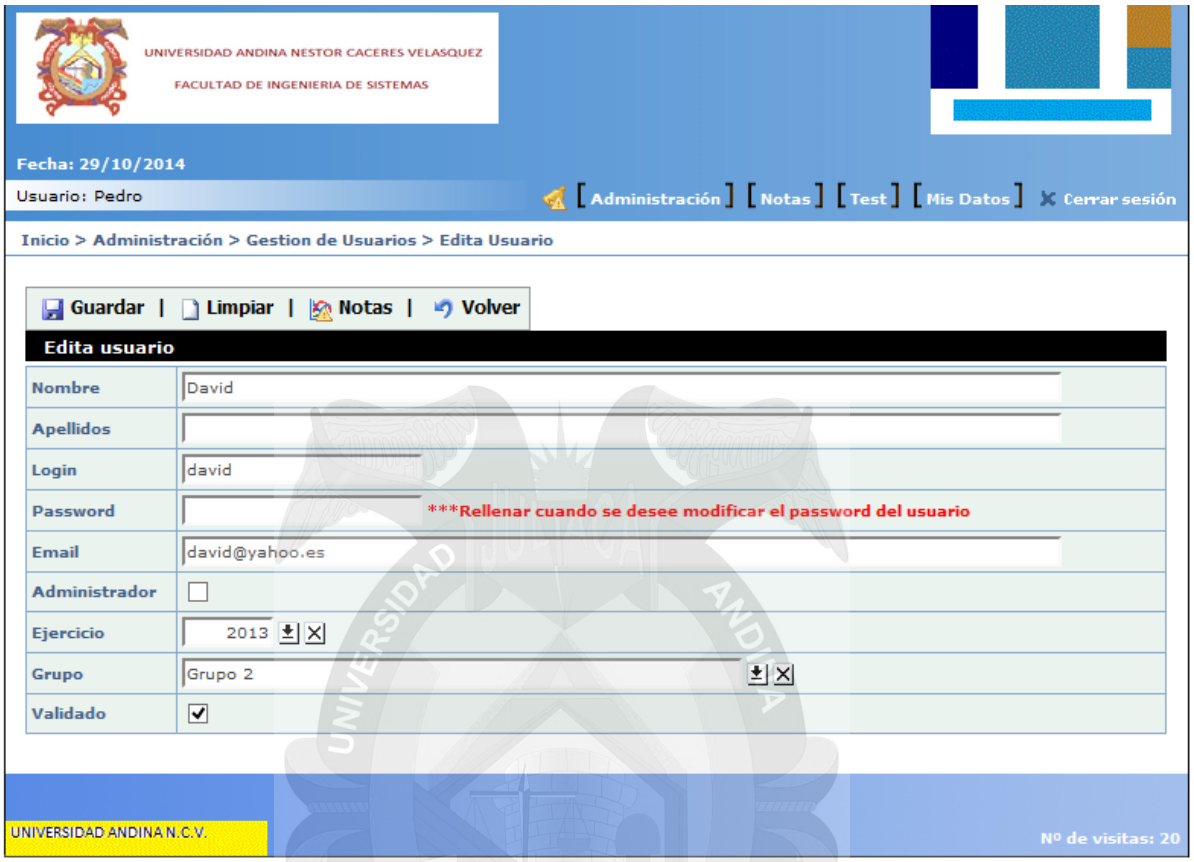
En la página de la figura *Fig 5.15* se puede observar la página que resulta al pulsar sobre el botón de lista de grupos de la página de la figura *Fig 5.13.* En dicha página vamos a poder cambiar la ordenación y para seleccionar un grupo sólo debemos pinchar en la fila del grupo que queremos coger.

Fig. 5.15.- Página donde se selecciona un grupo

En la siguiente página el administrador podrá dar de alta un usuario o modificar los datos de un usuario con sólo rellenar los campos y pulsar sobre el botón *Guardar*. Los campos ejercicio y grupo se rellena como pudimos ver en la página de la figura *Fig 5.13*.

Cuando estemos editando un usuario, tendremos activo el botón *Notas*, el cual, al pinchar sobre el, nos llevará a una página donde podremos ver todas las notas de los intentos realizado por el usuario (ver figura *Fig 5.17*).

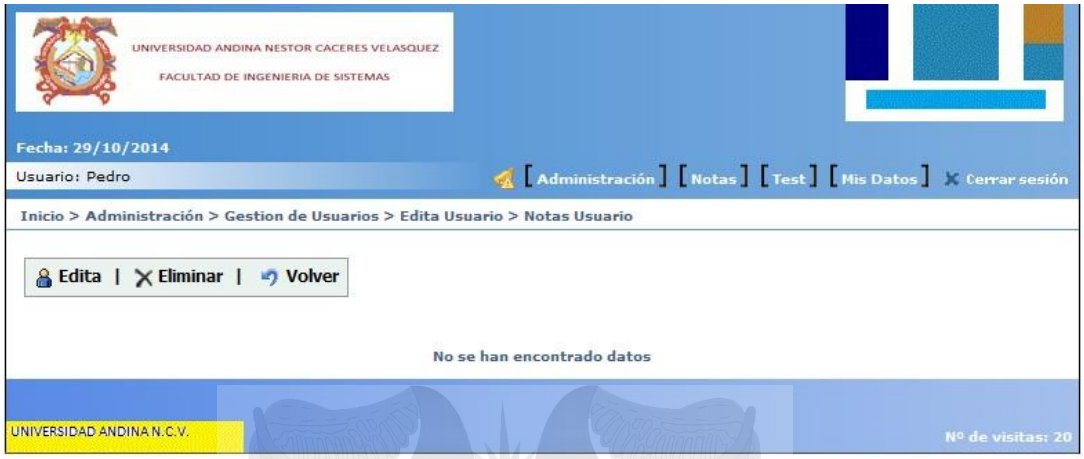
Fig. 5.15.- Página donde se inserta o se modifica los datos de un usuario



En la página siguiente vamos a poder paginar, cambiar la ordenación, eliminar registros, cambiar el número de registros mostrados por página, etc, como podíamos hacer en la página de gestión de usuarios.

Disponemos de un botón *Edita*, mediante el cual volvemos a los datos del usuario que estamos viendo.

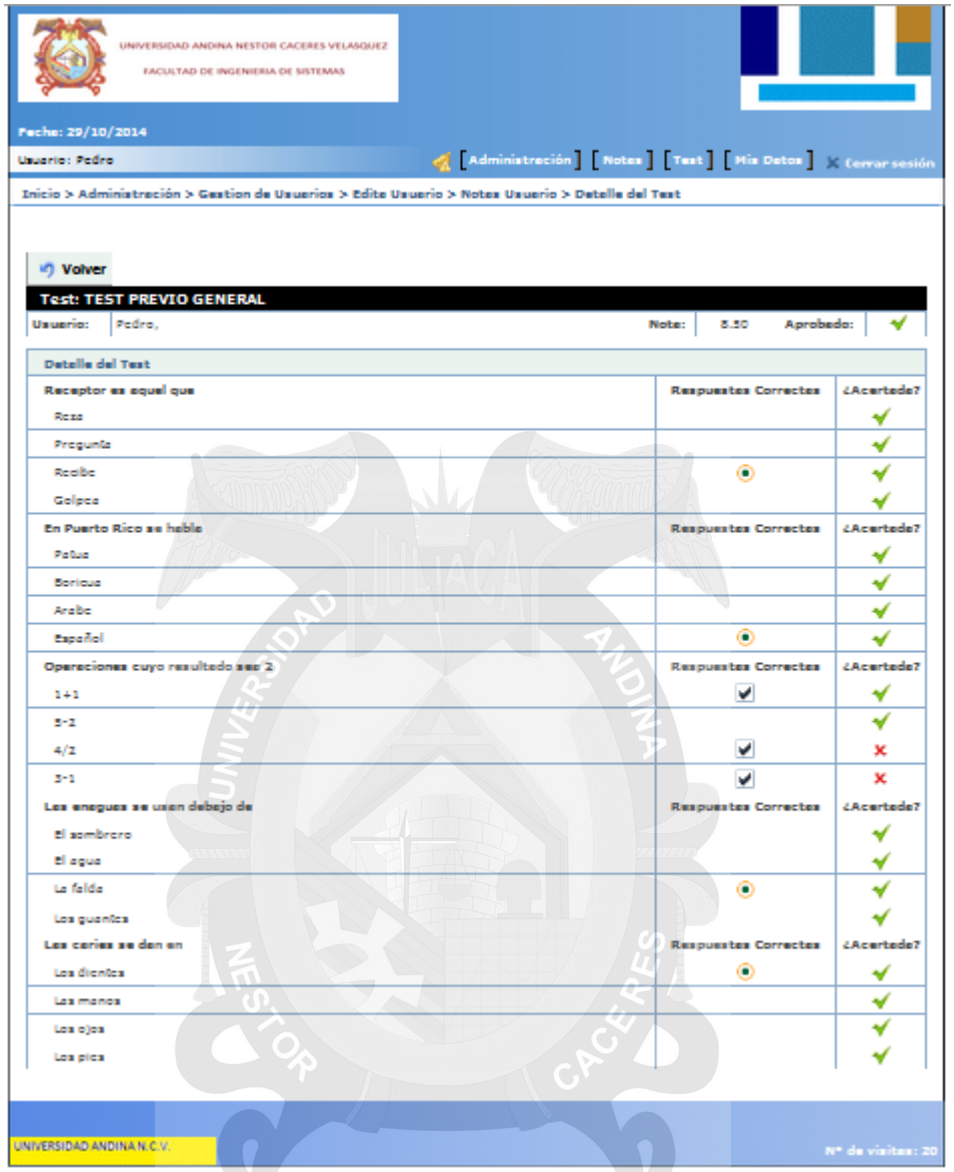
Fig. 5.17.- Página donde se muestra todas las notas de un usuario



Para ver el detalle de un Test, tendremos que pinchar sobre la imagen de la "lupa" de la figura *Fig 5.17*, la cual, nos llevara a la página de la figura *Fig 5.18*.

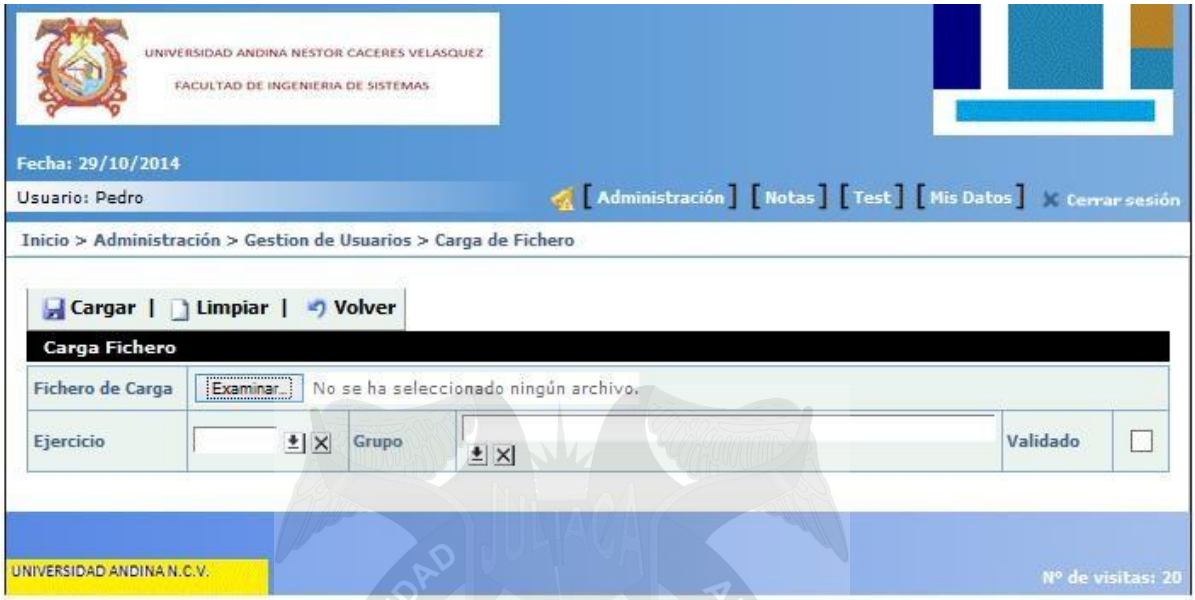
En la página de la figura *Fig 5.18*, podemos observar el alumno que ha realizado el test, el test realizado, así como la nota obtenida y si se ha aprobado o suspendido. También, se puede observar las preguntas que compusieron el test y sus respuestas, marcando cuales de estas eran las correctas y cuáles fueron las acertadas por el alumno.

Fig. 5.18.- Página donde se observa el detalle de un test realizado por un alumno



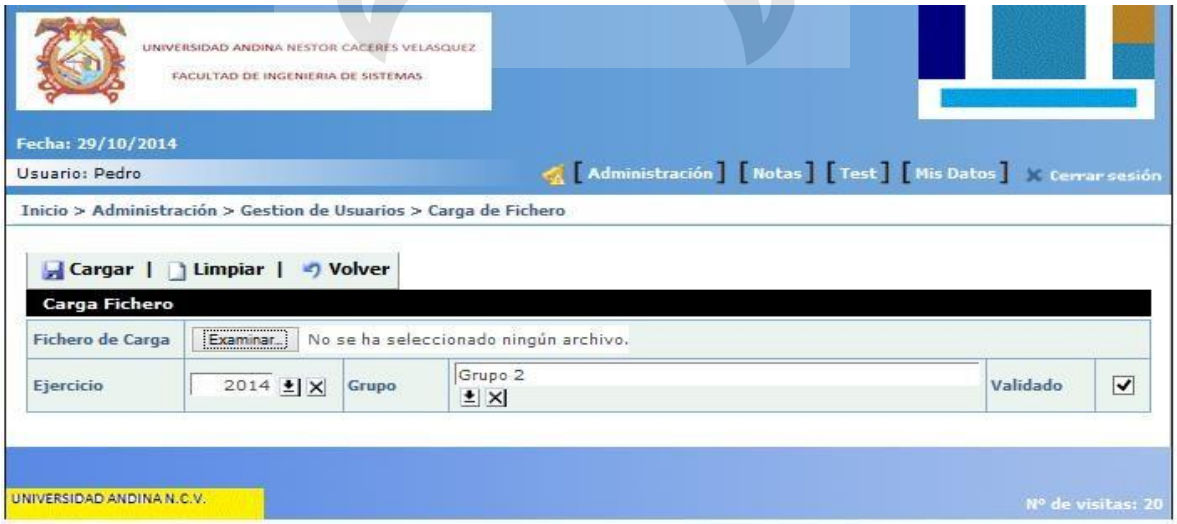
Una vez acabada la explicación de la *Gestión de Usuarios*, paso a detallar la opción de *Carga de Fichero*, mediante la cual, el administrador podrá introducir en la aplicación de forma fácil y rápida todos los usuarios (ver figura *Fig 5.19*).

Fig. 5.19.- Página donde se carga de forma automática los usuarios



Para cargar de forma automática una serie de alumnos, basta con seleccionar el fichero donde se encuentra la información de esos alumnos, seleccionar el ejercicio y el grupo al que van a estar asociados y si van a estar validados o no. Una vez seleccionado los campos basta con pinchar sobre el botón Cargar, el cual introduce los datos de los usuarios en la aplicación y muestra la información de cómo ha ido la carga, ver figura *Fig 5.20*.

Fig. 5.20.- Ejemplo de carga automática de usuarios



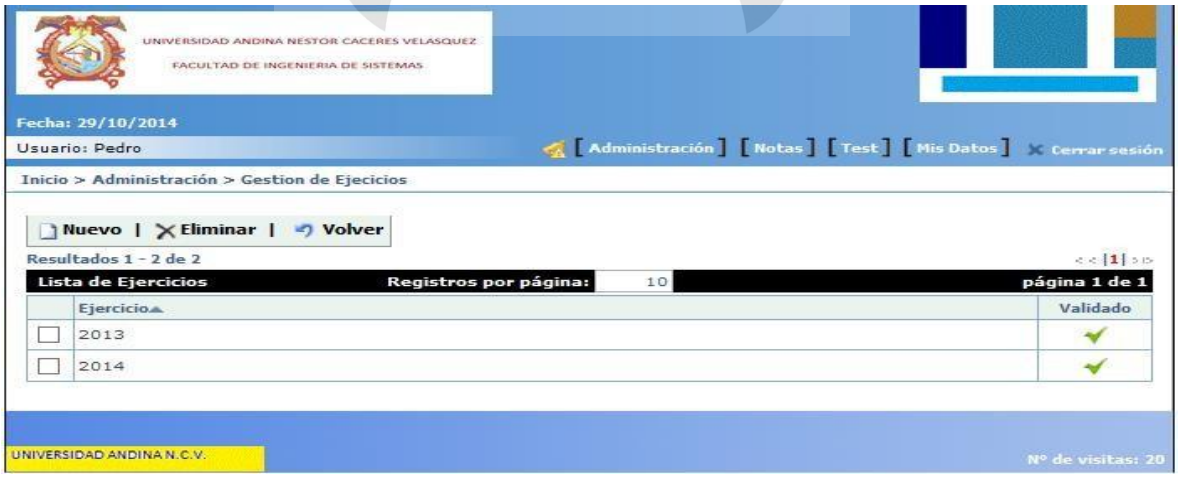
Una vez acabada la explicación de la *Carga de Fichero*, paso a detallar la opción de *Gestión de Ejercicios*, mediante la cual, el administrador podrá gestionar los ejercicios donde vamos a poder asignar alumnos (ver figura *Fig 5.21*).

Cuando entremos en la página nos mostrara un listado con todos los ejercicios que tengamos dados de alta y podremos ir navegando por las diferentes páginas para ver el resto de registros, incluso podremos cambiar la ordenación o el número de registros que se muestra en cada pantalla.

El administrador va a poder eliminar un ejercicio con sólo seleccionarlo mediante el check de la izquierda y luego pinchando sobre el botón *Eliminar*, siempre y cuando el sistema no encuentre dependencias, es decir no tengas elementos que dependa de este ejercicio, como puede ser alumnos que estén asignados a dicho registro.

Para editar un ejercicio basta con pinchar sobre cualquier campo del ejercicio en cuestión, lo veremos detallado en la figura *Fig 5.22* y para dar un ejercicio de alta, tendremos que pinchar sobre el botón *Nuevo* que nos llevará a la página de la figura *Fig 5.22* pero con los campos en blanco.

Fig. 5.21.- Página donde se gestiona los ejercicios



En la siguiente página el administrador podrá dar de alta un ejercicio o modificar los datos de un ejercicio con sólo rellenar los campos y pulsar sobre el botón *Guardar*.

Fig. 5.22.- Página donde se inserta o se modifica los datos de un ejercicio

Una vez acabada la explicación de la *Gestión de Ejercicios*, paso a detallar la opción de *Gestión de Grupos*, mediante la cual, el administrador podrá gestionar los grupos donde vamos a poder asignar alumnos (ver figura *Fig 5.23*).

Cuando entremos en la página nos mostrara un listado con todos los grupos que tengamos dados de alta y podremos ir navegando por las diferentes páginas para ver el resto de registros, incluso podremos cambiar la ordenación o el número de registros que se muestra en cada pantalla.

El administrador va a poder eliminar un grupo con sólo seleccionarlo mediante el check de la izquierda y luego pinchando sobre el botón *Eliminar*, siempre y cuando el sistema no encuentre dependencias, es decir no tengas

elementos que dependa de este ejercicio, como puede ser alumnos que estén asignados a dicho registro.

Para editar un grupo basta con pinchar sobre cualquier campo del grupo en cuestión, lo veremos detallado en la figura *Fig 5.24* y para dar un grupo de alta, tendremos que pinchar sobre el botón *Nuevo* que nos llevará a la página de la figura *Fig 5.24* pero con los campos en blanco.

Fig. 5.23.- Página donde se gestiona los grupos

En la siguiente página el administrador podrá dar de alta un grupo o modificar los datos de este con sólo rellenar los campos y pulsar sobre el botón *Guardar*.

Fig. 5.24.- Página donde se inserta o se modifica los datos de un grupo

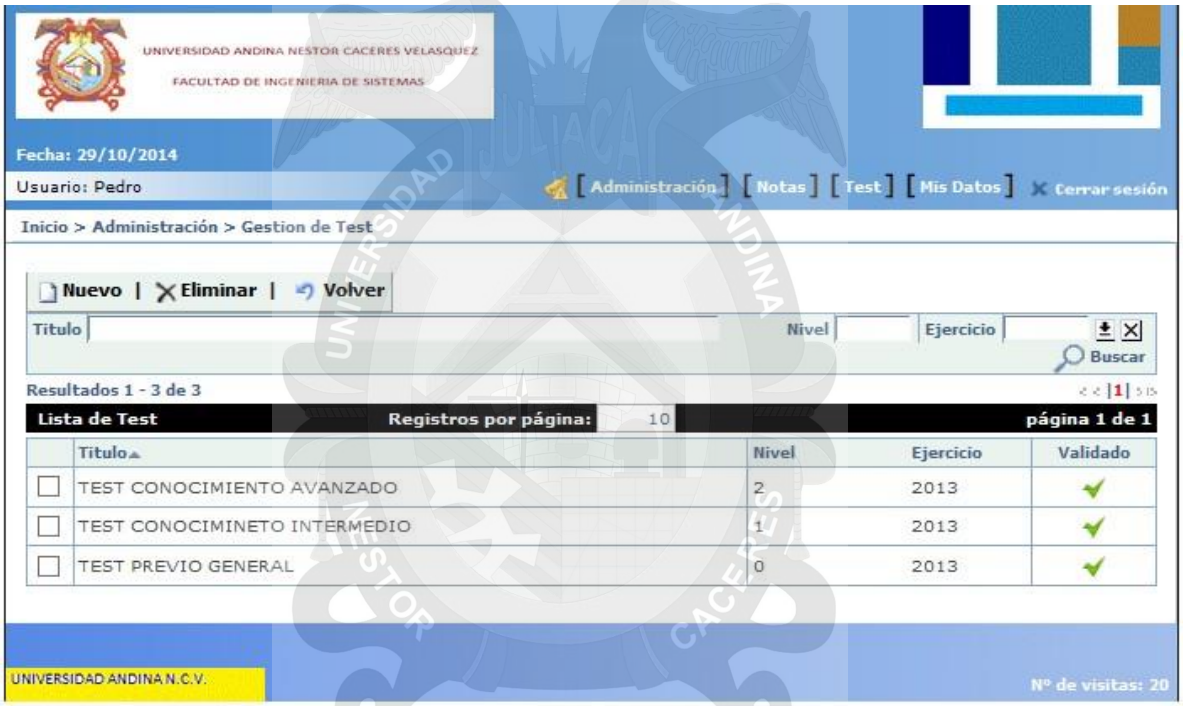
Una vez acabada la explicación de la *Gestión de Grupos*, paso a detallar la opción de *Gestión de Test*, mediante la cual, el administrador podrá gestionar los test que van a poder realizar los alumnos (ver figura *Fig 5.25*).

Cuando entremos en la página nos mostrara un listado con todos los test que tengamos dados de alta y podremos ir navegando por las diferentes páginas para ver el resto de registros, incluso podremos cambiar la ordenación o el número de registros que se muestra en cada pantalla.

El administrador va a poder eliminar un test sólo seleccionarlo mediante el check de la izquierda y luego pinchando sobre el botón *Eliminar*, siempre y cuando el sistema no encuentre dependencias, es decir no tengas elementos que dependa de este test, como puede ser notas de los intentos realizado por los alumnos que estén asignados a dicho registro. Al borrar un test, se borrarán todas las preguntas y todas las respuestas que tengan asociadas dicho test.

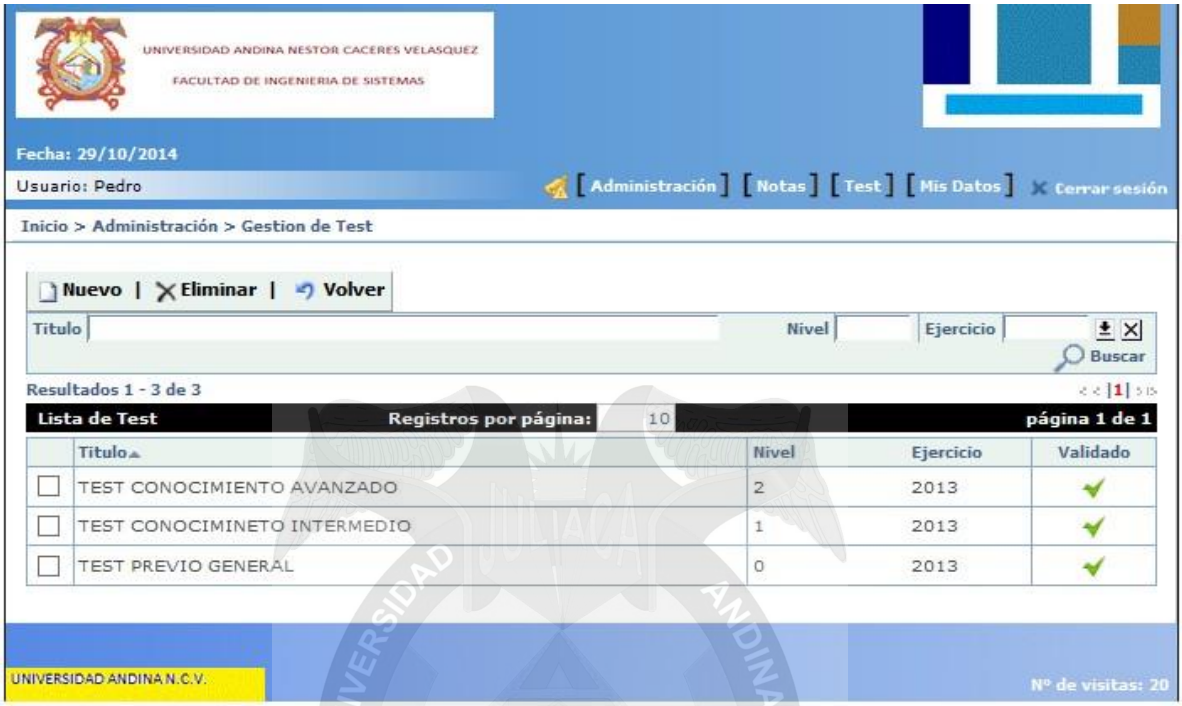
Para editar un test basta con pinchar sobre cualquier campo del grupo en cuestión, lo veremos detallado en la figura *Fig 5.25* y para dar un test de alta, tendremos que pinchar sobre el botón *Nuevo* que nos llevará a la página de la figura *Fig 5.26* pero con los campos en blanco y sin tener habilitada la opción de ver el detalle del test y ver todas las notas de los intentos realizados por los alumnos.

Fig. 5.25.- Página donde se gestiona los test



En la siguiente página el administrador podrá dar de alta un test o modificar los datos de este con sólo rellenar los campos y pulsar sobre el botón *Guardar*, siempre y cuando el test no haya sido realizado por algún alumno, caso en el que el test no podrá ser modificado ni borrado hasta que se eliminen esos intentos de realización.

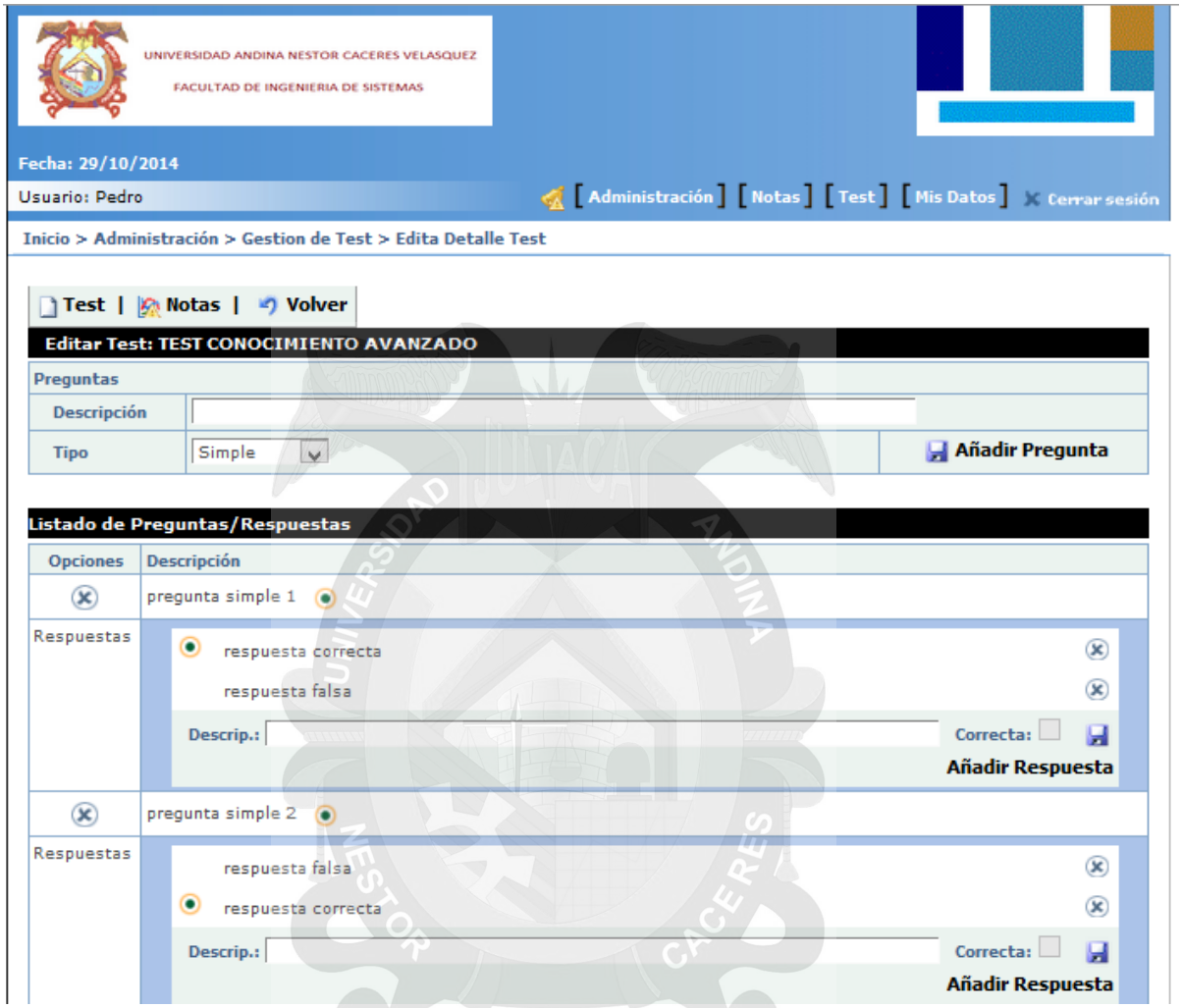
Fig. 5.26.- Página donde se inserta o se modifica los datos de un test



Si el administrador pulsa sobre el botón *Detalle* de la figura *Fig 5.26*, podrá ver la pantalla de la figura *Fig 5.27*, donde podrá dar de alta las preguntas y sus respectivas respuestas, las cuales forman el repositorio de preguntas y respuestas, de donde el sistema cogerá los datos para generar de forma dinámica los test, cuando es solicitado por el alumno.

Además, le damos al administrador la opción de volver a los datos del test, con sólo pinchar sobre el botón *Test*.

Fig. 5.27.- Página donde se inserta o se modifica las preguntas y respuestas de un test

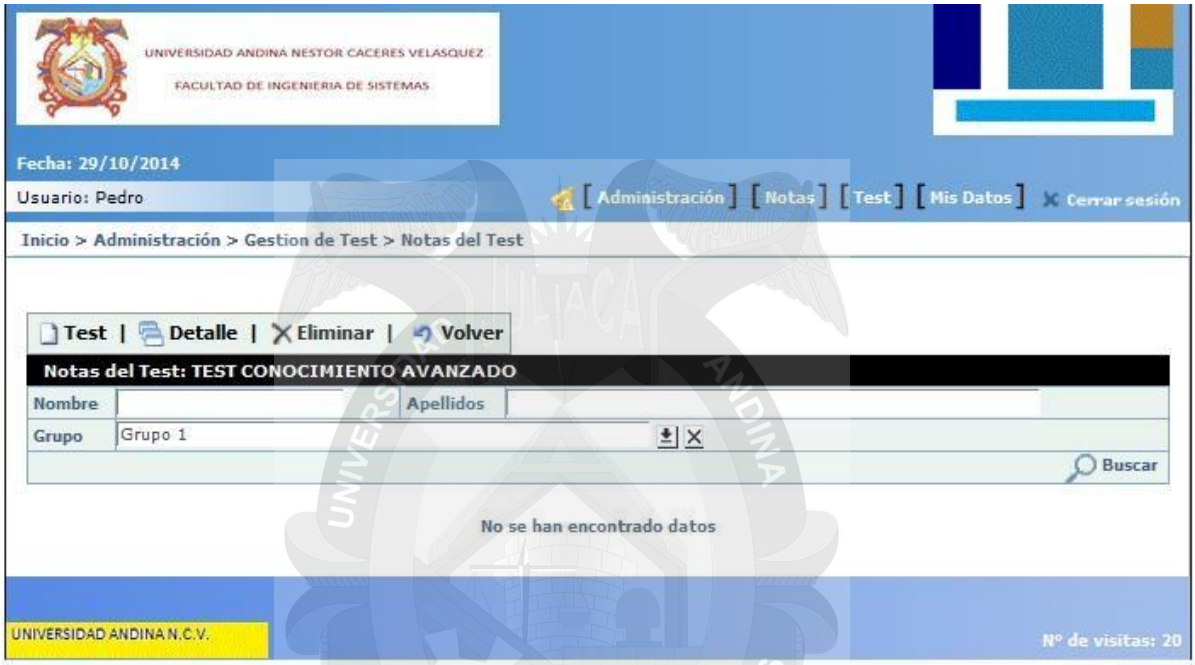


Si el administrador pulsa sobre el botón *Notas* de la figura *Fig 5.25*, podrá ver la pantalla de la figura *Fig 5.28*, donde podrá ver de forma rápida todas las notas de los intentos realizados por los alumnos.

Además, le damos al administrador la opción de volver a los datos del test, con sólo pinchar sobre el botón *Test* y de volver a las preguntas que lo forma con pinchar sobre el botón *Detalle*.

En la página siguiente el administrador podrá filtra los datos mediante una búsqueda, podrá paginar, cambiar la ordenación e incluso podrá borrar algún intento, si fuera conveniente.

Fig. 5.28.- Página donde se gestiona los intentos realizado de un test



Para ver el detalle de un Test, tendremos que pinchar sobre la imagen de la "lupa" de la figura *Fig 5.28*, la cual, nos llevara a la página de la figura *Fig 5.18*, la cual, se detalló anteriormente.

Conclusiones

1. Se implementó una plataforma web que permite realizar evaluaciones en línea automatizando este proceso, con lo cual se cumplió con lo propuesto en los objetivos del proyecto.
2. Se empleó las técnicas de Ingeniería WEB para el desarrollo de este proyecto con lo cual se logró administrar el proceso de desarrollo de software en forma eficiente.
3. Se empleó software libre para el desarrollo de la aplicación con lo cual se demuestra que el desarrollo de este tipo de aplicaciones es bastante bueno el uso de estas herramientas ya que proveen la potencia necesaria para su creación.
4. Se logró disminuir el tiempo de elaboración y calificación de exámenes, lo que se hizo posible con el sistema Web creado ya que automatiza este proceso de creación y corrección del mismo.

Recomendaciones

Se recomienda continuar con el desarrollo de este proyecto implementando los módulos que corresponden para ser incluido en un sistema académico completo que controle un sistema de evaluaciones completo.

Se recomienda configurar un servidor WEB que permita instalar y utilizar herramientas WEB en la Facultad de Ingeniería de Sistemas de la UANCV, con el fin de poder bridar estos servicios a los estudiantes y egresados de la facultad con el fin de realizar este tipo de trabajos.

Se recomienda incentivar el uso de las TIC's en la facultad de ingeniería de sistemas para asi poder mejorar la enseñanza de los diferentes cursos en esta facultad.

Bibliografía

Fisher, J. M. (1997). *El Libro del WEBMASTER.* España: Anaya Multimedia.

Klander, L. (1998). *A Prueba de Hackers.* España: Anaya Multimedia.

Martinez, R. (2002). *Maual de PHP.* España: Anaya.

ORACLE. (2013). *MySql Manual de Referencia.* ORACLE.

PHP. (s.f.). *La WEb de PHP*. Obtenido de PHP. Net: www.php.net

Platt., E. S. (1997). *HTML 4.0.* ESPAÑA, ESPAÑA: ANAYA MULTIMEDIA.

- **Información obtenida a través de Internet.**

- *Direcciones de Páginas Web.*

  www.php.net
  www.php-es.com
  www.apache.org
  http://httpd.apache.org
  www.programacion.net/php/mysql.html
  http://es.tldp.org/Manuales-LuCAS/manual_PHP/manual_PHP/
  www.easyphp.org
  www.mysql.com